

MONTEREY CALIF 4840
* JAL * MONTEREY *
* JAL * MONTEREY *
* JAL * MONTEREY *

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

THE ENRICHMENT OF SMOLER'S
MODEL OF LAND COMBAT

by

Glenn M. Mills

September 1980

Thesis Advisor:

J. G. Taylor

Approved for public release; distribution unlimited

T195910

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

1. REPORT NUMBER		2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) The Enrichment of Smoler's Model of Land Combat		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; September 1980	
7. AUTHOR(s) Glenn M. Mills		6. PERFORMING ORG. REPORT NUMBER	
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)	
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE September 1980	
		13. NUMBER OF PAGES 82	
		15. SECURITY CLASS. (of this report) Unclassified	
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.			
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)			
18. SUPPLEMENTARY NOTES			
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Lanchester-type ground combat model.			
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis provides the student of combat modelling with a computer program for a relatively simple combat model that can be used in a classroom environment for study and analysis. The model is an aggregated, force-on-force ground-combat model that uses			

UNCLASSIFIED

Lanchester's aimed-fire equations for casualty assessment. The original version of the model was developed in 1979 in a previous thesis, and shortfalls in the original model have been overcome, along with the addition of several enrichments to provide added user flexibility. A user's manual is provided to facilitate user access to the model from a permanent disk in the W. R. Church Computer Center.

Approved for public release; distribution unlimited

The Enrichment of Smoler's
Model of Land Combat

by

Glenn M. Mills
Captain, United States Army
B.S., North Carolina State University, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OPERATIONS RESEARCH

from the

NAVAL POSTGRADUATE SCHOOL
September 1980

ABSTRACT

This thesis provides the student of combat modelling with a computer program for a relatively simple combat model that can be used in a classroom environment for study and analysis. The model is an aggregated, force-on-force ground-combat model that uses Lanchester's aimed-fire equations for casualty assessment. The original version of the model was developed in 1979 in a previous thesis, and shortfalls in the original model have been overcome, along with the addition of several enrichments to provide added user flexibility. A user's manual is provided to facilitate user access to the model from a permanent disk in the W. R. Church Computer Center.

TABLE OF CONTENTS

I.	INTRODUCTION -----	6
II.	THE ORIGINAL MODEL -----	9
	A. GENERAL DESCRIPTION -----	9
	B. ATTRITION PROCESS -----	9
	C. UNIT LOCATION AND MOVEMENT -----	11
	D. BATTLE TERMINATION -----	12
	E. DETECTION AND FIRE ALLOCATION -----	13
III.	MODEL CHANGES -----	15
	A. GENERAL -----	15
	B. CONDITIONALLY-DETERMINISTIC-ATTRITION -----	15
	C. UNIT LOCATION AND MOVEMENT -----	20
	1. Straight Line Routes -----	21
	2. User Determined Routes -----	21
	D. BATTLE TERMINATION -----	23
	E. ALTERNATE DEFENSIVE POSITIONS -----	25
IV.	FUTURE ENHANCEMENT -----	27
	A. HETEROGENEOUS FORCES -----	27
	B. AMMUNITION EXPENDITURE -----	28
	C. ARTILLERY -----	29
APPENDIX A:	USER'S MANUAL -----	31
APPENDIX B:	PROGRAM LISTING -----	49
APPENDIX C:	DEFINITION OF VARIABLES IN COMPUTER PROGRAM-----	77
BIBLIOGRAPHY	-----	81
INITIAL DISTRIBUTION LIST	-----	82

I. INTRODUCTION

As the art of combat modelling becomes more advanced, combat modelers are continually building more and more complicated models. To the beginning modeler, the ability to understand how these models operate is difficult, if not impossible. As a student of combat modelling, I have sensed a need for a relatively simple model that could be easily studied in a classroom environment. It was with this in mind that this project was initiated.

At present, there seems to be no simple combat model available that demonstrates the basics of model building to the beginning student. In 1979, J. Smoler (a student at the Naval Postgraduate School) attempted to build such a model for his thesis research (Smoler, 1979). His model was a deterministic, force-on-force computer model that used Lanchester's aimed-fire equations for casualty assessment. The general scheme of his model is shown in Figure 1.

Although Smoler's model was a bold attempt at a simplistic combat model, it did have some problem areas that warranted investigation. Some of these problems were first discovered during a class project in a combat models class (OA 4655) at the Naval Postgraduate School. It is the purpose of this thesis to study Smoler's model and to undertake an enrichment program that will make the model

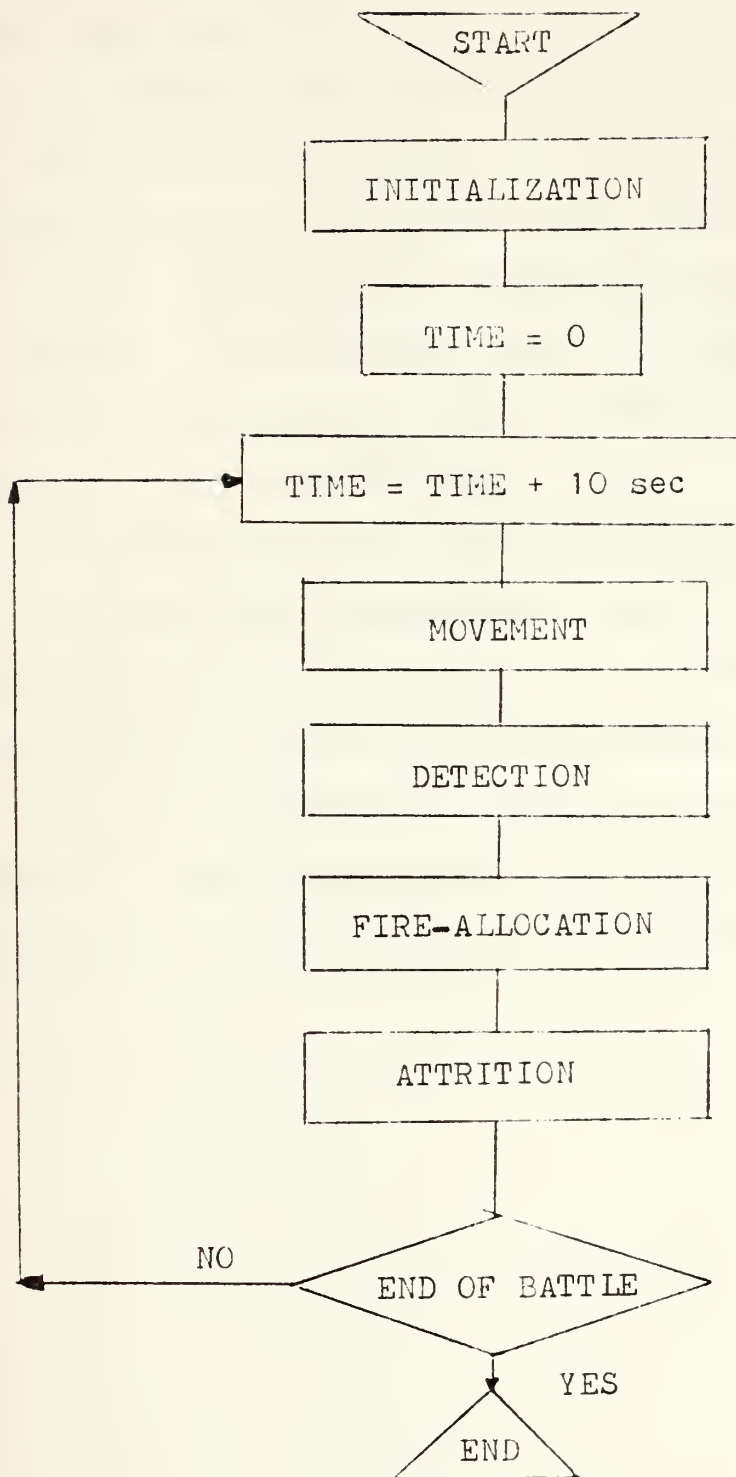


FIGURE 1. GENERAL SCHEME

more realistic and flexible, while maintaining it's transparency and simplicity.

Once enhanced, the model would then be made available for use as an instructional tool for combat modelling classes. This will be accomplished through the development of a user's manual to facilitate the use of the model by students by placing the model on a permanent disk in the W. R. Church Computer Center where it will be easily accessed by any desired user.

The remainder of this paper will discuss, in detail, the problems found with Smoler's model and the methods used to solve these problems. It will also outline some new features that have been incorporated into the model to allow user flexibility. Finally, the User's Manual (Appendix A) will provide all the required information to enable even the novice user to utilize the model.

II. THE ORIGINAL MODEL

A. GENERAL DESCRIPTION

Smoler's original model of land combat is a deterministic model that plays combat between two homogeneous forces, a blue force and a red force. The blue force is comprised of three subunits in a static defense, with each subunit armed with three TOW antitank missile systems. The red force is composed of three subunits of three tanks each, attacking on pre-planned routes. The battle takes place on the 10 x 10 Km Fulda Box that has been developed and used in the STAR simulation model (Wallace, 1978). Since the major thrust of this paper is to alter the original model to a form that will be easily used for classroom instruction, a brief discussion of the major components of the original model will be presented, including problem areas that have motivated model changes.

B. ATTRITION PROCESS

The attrition process in the original model utilizes Lanchester "aimed-fire" equations with variable attrition coefficients. The Lanchester equations used are fairly simple and will be discussed later. However, the calculation of the attrition coefficients is of more immediate interest. Smoler used the Bonder-Farrell model to calculate the coefficients, A_{ij} , the rate at which one firer

of unit i kills unit j targets. These A_{ij} 's are computed according to:

$$A_{ij} = 1/E(T_{ij})$$

where $E(T_{ij})$ is the expected time for one firer of unit i to kill one target of unit j. The $E(T_{ij})$ is computed using the Bonder-Farrell formula:

$$E(T_{ij}) = t_a + t_l - t_h + (t_h + t_f)/P(K|H) + \\ (t_m + t_f)/P(h|m) \times ((1-P(h|h))/P(K|H) + \\ P(h|h) - P)$$

where

t_a = time to acquire a target

t_l = time to fire first round following acquisition

t_h = time to fire following a hit

t_m = time to fire following a miss

t_f = time of flight of a round

P = probability of a first round hit

$P(h|h)$ = probability of a hit following a hit

$P(h|m)$ = probability of a hit following a miss

$P(K|H)$ = probability of a kill given a hit

This formula holds for the conditions that the hit probability of any round depends only on the result of the previous round and no accumulated damage is considered.

Smoler also assumed that for the TOW weapon system,

$P(K|H) = 1.0$ and $P(h|m) = P(h|h) = P$, thus reducing the

equation to:

$$E(T_{ij}) = t_a + t_l + t_f + (t_m + t_f)(1-P)/P$$

Smoler also assumed that for tanks, $P(K|H) = 1.0$ and $t_f = 0$, reducing the formula to:

$$E(T_{ij}) = t_a + t_l + t_m(1-P)/P(h|m)$$

Utilizing these equations to calculate the A_{ij} 's, the attrition during each time step was computed using the Euler-Cauchy differencing equations to approximate Lancaster's force-on-force attrition differential equations.

C. UNIT LOCATION AND MOVEMENT

Red and blue unit locations were handled in two different ways in the original model. The blue locations were left as user inputs, while the red locations had been pre-determined by the model builder and could not be altered by the user. This allowed flexibility of defensive positions, but because of the method of determining movement routes for the attackers, this flexibility was limited.

The method utilized for route determination was, at best, unrealistic. For each original red location, a straight west to east route was calculated. Each route was divided into 40 meter intervals, since each red unit was assumed to move that distance during each 10 second time interval. This approximated an average rate of movement of 9 m.p.h. This method of route selection, although easy to implement, has several significant shortcomings. It does not allow the attacking units to utilize terrain

features during movement. Also, it permits attackers to move over terrain that is, in real life, impassible.

D. BATTLE TERMINATION

In any combat model, adequate battle termination rules must be considered. Smoler utilized two criteria to terminate the battle. The first of these was annihilation of one of the two forces. This criteria is reasonable in an expected value model like this and was adequately handled. This is not the case for the second termination rule.

The second rule for termination is that the distance between red and blue forces becomes too small. The use of this criteria for a stopping rule is fine, provided it is implemented properly. In the original model, a geographic center was calculated for each force. It was these center of mass points that were compared to determine if the units were too close. This method can lead to problems, such as those that were encountered during some initial trial runs. In one battle, the red units, during their advance, had eliminated one entire blue unit and had suffered the loss of one of their own units. Since the red units are allowed to advance as much as 150 meters apart and blue units were set up so they were approximately 1300 meters apart, there was considerable distance between units on the battlefield. Since the two units had been destroyed, the center of mass computation allowed the red

units to move completely through the blue defense, while both sides still had forces remaining to fight with. This reflects a problem with the center of mass computation as a method for determining distances between units.

E. DETECTION AND FIRE ALLOCATION

Both detection and fire allocation processes are handled well in the original model and no changes have been made. A detailed description of these methods is contained in Smoler's Thesis (Smoler, 1979). However, a brief and general discussion of both processes will be included here.

The detection phenomena is modeled in two ways. First, a non-firing detection can occur as a result of an observer's random search within his designated section of responsibility. A 30° field of vision for an observer is assumed, and the probability that an observer is looking in the direction which enables him to detect a target is computed by integrating the Limicon Function over limits that are $\pm 15^\circ$ from the primary direction the observer is looking. The Limicon Function, $f(\theta)$, is the following probability density function:

$$f(\theta) = A + B \times \cos(\theta)$$

where

D = assigned sector width/2

$A = -B \times \cos(D)$

$B = 1 / 2(\sin(D) - D \times \cos(D))$

θ = primary direction observer is looking

Also, A and B are chosen such that

$$\int_{-D}^D f(\theta) d\theta = 1$$

The second method of detection is a firing detection. This phenomena occurs when the following happens. If a firer's location is within $\pm 15^{\circ}$ of an observer's primary direction of observation when he is firing, he is assumed to be detected and is added to the observer's target list. This models the detection of a firer by locating a firing signature of a weapon.

The fire allocation process is also modeled well by Smoler. Since each firing unit is not restricted to firing at one target, a fraction of each firing unit is allocated to each target on the firer's target list. This fraction is determined as a function of range to each target and of predetermined priorities of fire that are set so each unit's fire priority goes to those targets to his immediate front. Targets to the firer's flanks are then allocated a smaller percent of the available firepower. Again, the detection and fire allocation processes have been handled well and no changes to these routines will be made.

III. MODEL CHANGES

A. GENERAL

As the previous section outlines, there were several major problems discovered in Smoler's original model. In order to make the model more flexible and classroom useable, several major changes were found to be necessary. This chapter will provide a detailed description of these changes, which include conditionally-deterministic attrition, unit locations and movement, battle termination and movement of defenders.

B. CONDITIONALLY-DETERMINISTIC ATTRITION

In order to introduce a stochastic process into the attrition computation for this model, several options were available. These include the use of a Markov-process to determine casualties or the use of random attrition-rate coefficients. After considering both options, and in keeping with the transparent nature of this model, it was decided that stochastic attrition coefficients should be utilized, because the attrition-rate coefficient is a random quantity measuring a unit's fighting ability and can be realized before any given battle. The following procedure was used to develop a method for random attrition-rate coefficients.

The attrition-rate coefficient, A_{ij} , is a measure of the rate a firer of unit i attrits a target of unit j .

This also can be interpreted as a measure of the fighting ability of an element of type i . It is intuitively obvious that this is a variable quantity that can be affected by many different factors, including weather, esprit-de-corps, previous engagements and leadership, to mention only a few. In a homogeneous model, one can attempt to capture these random effects by developing a distribution of initial fighting unit capabilities, i.e., initial A_{ij} 's for each unit. Since no data is readily available that captures this phenomena, an attempt was made to fit a distribution to an intuitive feeling as to how this fighting ability varies from one unit to the next prior to a given engagement. It was the author's intuitive feeling that the A_{ij} 's should be distributed between .3 and .8 with the majority of the units being rated between .5 and .6.

Utilizing the above intuition, an attempt was made to "fit" a distribution to these assumed fighting levels. Graphically, the distribution would look like Figure 2.

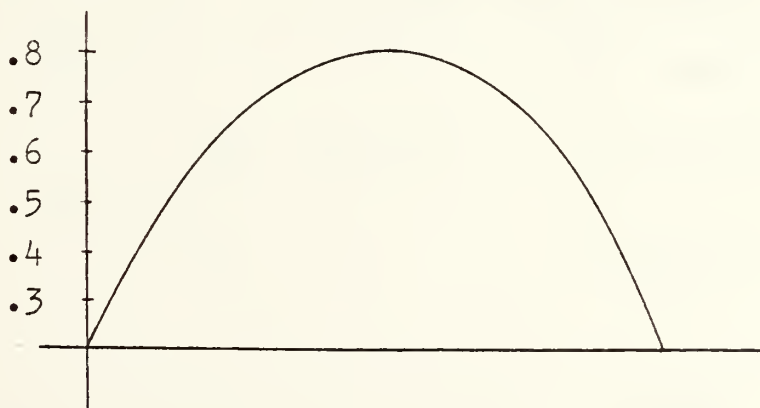


Figure 2. Distribution of Initial A_{ij} 's

Initially, an attempt was made to fit a Beta distribution to this curve. The fit was fairly accurate in the middle range, but was not satisfactory in the tails. The same was true for a Normal approximation. It was finally determined that the best fit was a straight quadratic fit, where a Uniform(0,1) input variable could be used and the output would yield the desired random number. The quadratic fit that was used is:

$$A_{ij}^0 = -2U^2 + 2U + .3$$

Once a distribution for the attrition-rate coefficients had been derived, the implementation into the model was accomplished. Since it was assumed that the fighting ability of each unit is a random quantity prior to a given battle, it was only necessary to obtain a realization of the random variable for each unit prior to the initiation of the battle. This realization, A_{ij}^0 , is determined by using a random Uniform(0,1) number and the above formula. A new attrition-rate coefficient for each unit is then computed during each time step using the equation:

$$A_{ij} = \begin{cases} A_{ij}^0 (1-r/r_e)^2 & \text{for } 0 \leq r \leq r_e \\ 0 & \text{for } r_e \leq r \end{cases}$$

where

r_e = maximum effective range of a firer's weapon

r = current range between firer and target

A_{ij}^0 = realization of unit's fighting ability

This equation is utilized because it varies with range, but it also is a function of A_{ij}^0 , thus creating a different attrition-rate curve for each unit, depending on that unit's fighting ability prior to the battle. Graphically, as an example, the attrition-rate coefficient curve for an A_{ij}^0 of .6 and a maximum effective range, r_e , of 3000 meters would look as in Figure 3.

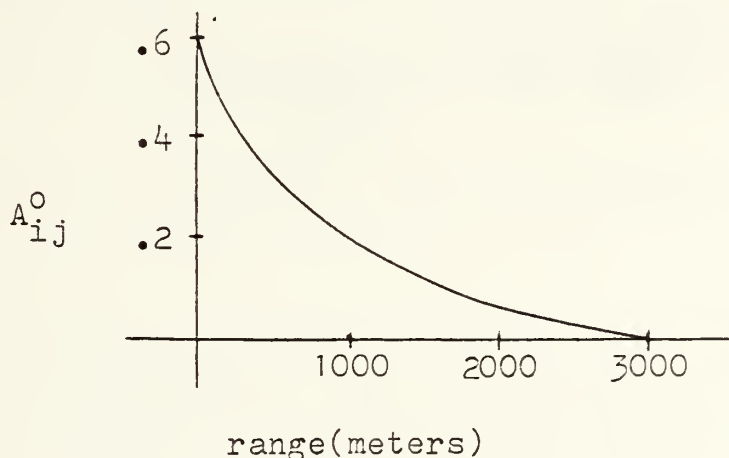


Figure 3. Attrition-rate coefficient curve for $A_{ij}^0 = .6$ and $r_e = 3000$ m.

Once the above method for determining attrition-rate coefficients had been selected, it was coded and included as a user option in the program. Once the code had been implemented, several runs were made utilizing different random number seeds to compare battle results using this method for attrition-rate coefficient determination. These runs were then compared with a run using the original deterministic method. The results are summarized below.

First, the model was run using the deterministic attrition method. For purposes of comparison, all locations,

movement rates, force levels and other input variables were held constant for all runs. Combatant attrition using the deterministic process as a function of time is shown in Figure 4.

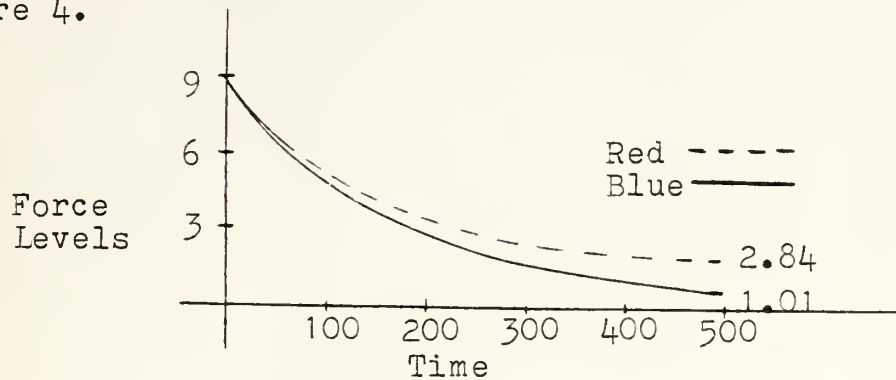


Figure 4. Attrition using deterministic attrition

In this battle, termination occurred as a result of forces being too close together, with force levels at termination reflected in Figure 4. This was then compared to the runs that utilized the new attrition module. The results of four of these runs are in Figure 5. The A_{ij}^0 's are the realizations of the random variable denoting a unit's initial fighting capability prior to the battle. The three numbers under the red and blue headings are the realizations for each subunit in the battle on which casualty assessments are computed.

These results show, in fact, that a randomly selected A_{ij}^0 does have an effect on the final outcome of the battle. This module now provides the user the option of selecting a conditionally-deterministic process that still uses Lanchester's equations as the basis for the attrition computation.

A_{ij}^O						Force Level			
Blue			Red			Blue	Red	Time	Result
.33	.77	.46	.79	.49	.56	1.7	1.5	780	Too close
.39	.67	.58	.63	.78	.36	4.3	0	330	Blue wins
.78	.67	.80	.74	.68	.72	3.2	0	320	Blue wins
.32	.43	.68	.51	.77	.65	1.2	3.8	640	Too close
Deterministic						1.1	2.8	460	Too close

Figure 5. Comparison of Stochastic and Deterministic Battles

C. UNIT LOCATION AND MOVEMENT

As was pointed out in an earlier section, the unit locations and movement logic that Smoler used tended to be rigid and inflexible. The changes that have been made to rectify this and allow more flexibility for the user will be discussed here.

First, the user is now responsible for inputting all combatant locations, both attackers and defenders. The format for these inputs will be specified in Appendix A, the User's Manual. Additionally, the user is given the option of selecting routes for the red advance or allowing the model to calculate straight west to east routes as in the past. To add more flexibility, the user will also be required to input the rate of advance (vehicle speed) for the attacking forces. The inputted rate of advance is used in both methods of route calculation, thus controlling the red attack.

1. Straight Line Routes

The option to use straight line routes from west to east is handled in the same manner as before. The route is broken down into discrete distance intervals from the initial red location straight to the east. The only difference is that the length of each interval is determined by the vehicle speed that the user inputs. The route for each red unit consists of 125 equal length intervals.

2. User Determined Routes

The option to allow the user to select routes for the red advance has been added to enable the user to make use of available terrain and to add realism to the model. The method for calculating these routes is straightforward and is discussed next.

The user is required to input the original location of each red subunit and from one to ten nodes he wishes each attacking subunit to move through. This information, along with vehicle speed, is used to calculate route intervals that move the attacking unit through each of the designated nodes. A complete route would look like the one depicted in Figure 6. The method used to compute the routes is as follows.

The straight line ground distance between the first two adjacent nodes, $DIST$, is calculated as shown in Figure 7. The angle between the desired direction of movement and straight west to east movement, a , is then calculated. Utilizing these quantities and the distance desired to

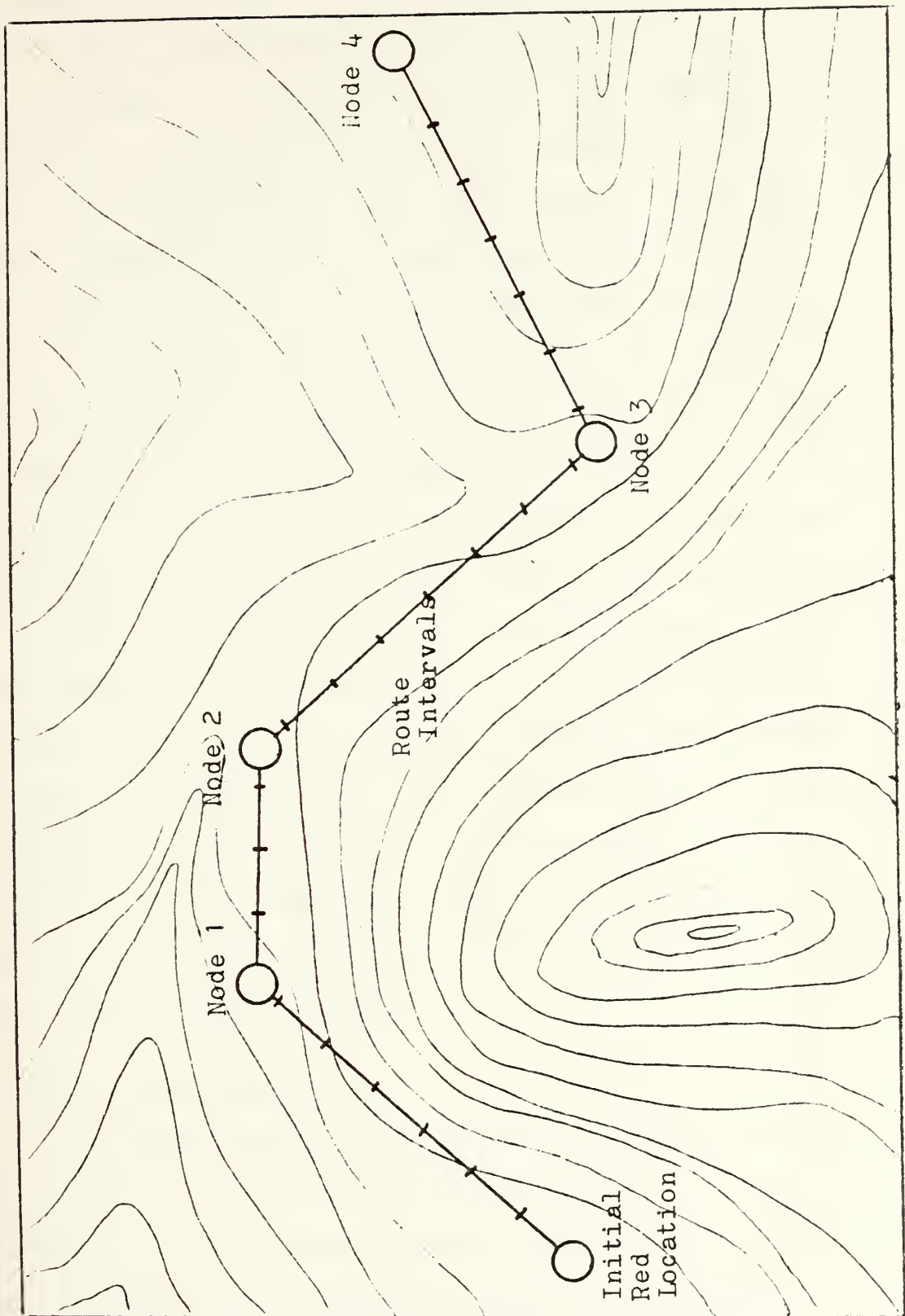


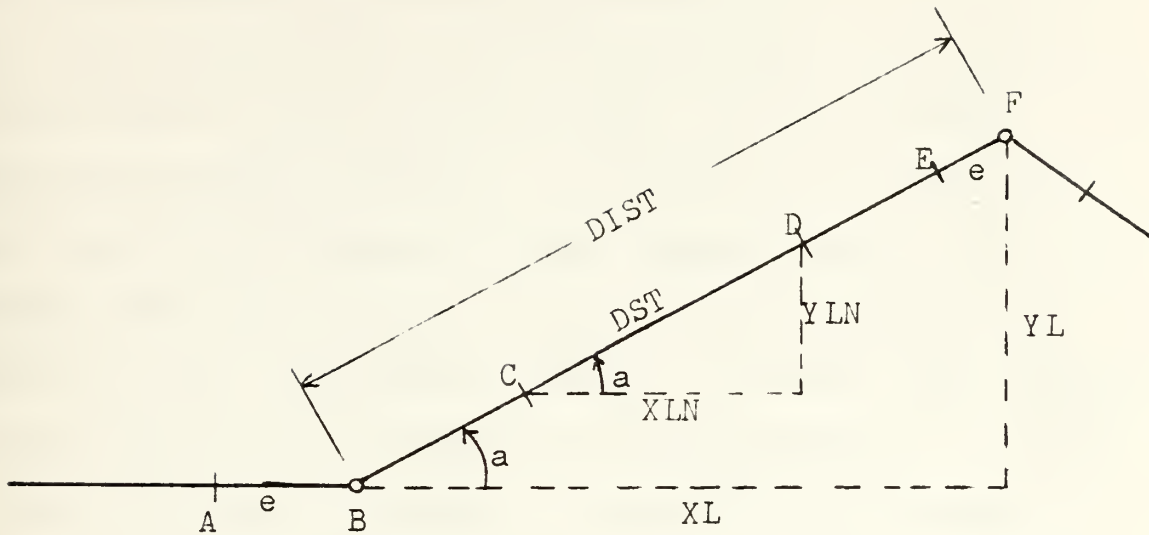
Figure 6. User Determined Routes

move during each time step, DST, the distance to be moved in the x and y direction, XLN and YLN, is now computed as shown in Figure 7. These distances are then added to the coordinates of the previous interval endpoint, point C in the figure, to determine the coordinates of the next interval endpoint, point D. This same distance is again added to compute the coordinates of the next endpoint, point E. This process is continued until the distance from the last endpoint computed to the next node is less than DST. This process is repeated between the next two nodes until the entire route is completed. In order to insure all intervals are of equal length, the computation of the first interval between any two nodes must be considered separately, by taking into account the distance left over from the last computation between the previous two nodes. To do this, the first interval takes the remaining distance, e, and adds it to an interval length of $DST - e$ for the first interval between any two nodes. This insures that each interval along the route is of length DST, which is the required length.

D. BATTLE TERMINATION

As previously stated, the battle is terminated either by annihilation of one of the two forces or by forces getting too close. The annihilation criteria has not been changed. However, to insure the distance criteria is effective, two changes have been made.

First, the distance between units is no longer calculated



NODE	COORDINATES	USE
A	XIC(N-1),YIC(N-1)	(N-1) st Interval Endpoint
B	XLOC(I,J)	User Inputed Node
C	XIC(N),YIC(N)	N th Interval Endpoint
D	XIC(N+1),YIC(N+1)	(N+1) st Interval Endpoint
E	XIC(N+2),YIC(N+2)	(N+2) nd Interval Endpoint
F	XLOC(I,J+1)	User Inputed Node

$$\text{DIST} = \sqrt{X_L^2 + Y_L^2}$$

$$a = \tan^{-1} (Y/XL), \text{ where } Y = |Y_L|$$

e = distance less than DST at end of calculation
of intervals between adjacent nodes.

$$Y_{LN} = DST \times \sin(a) \qquad X_{LN} = DST \times \cos(a)$$

Figure 7. Route Computation

as a center of mass distance. Instead, a distance is computed between each attacking subunit on which casualties are being computed that is still alive and each defending subunit that is still in the battle. If any of these distances becomes too small, the battle is considered to have moved to a "close-in" combat (hand-to-hand) that this program does not currently model. For this reason, the battle is then terminated. However, to insure the attackers do not pass by the flanks of remaining defenders and remain outside termination distance, a check of x coordinates for each subunit is also made. If any attacking subunit's x coordinate places him past the location of the forward most blue defensive subunit still in the battle, the battle is also terminated.

The criteria for being too close has been left as a user determination. It is one of the inputs that is described in the User's Manual, Appendix A. This allows for flexibility of breakpoint distances which also lends itself to the study of optimum breaking distances for various weapon systems on the battlefield.

E. ALTERNATE DEFENSIVE POSITIONS

The last addition to the model, added to increase the flexibility and realism of the battle, is the option of having the defending units move to alternate positions if the attackers close to within a user specified range. The move is handled in a simple and transparent manner.

If the user selects this option, the defender will move

when the attacking forces close within the breakpoint distance. When this occurs, each defensive unit that is still alive will move to an alternate position the user has selected. The duration of the move is also a user input. He simply specifies the number of 10 second time steps he wants to allow the blue forces to move. At the completion of the move, during which the red forces continue to attack, the battle will continue until one of the termination criteria discussed above has been met.

IV. FUTURE ENHANCEMENT

Although the above changes have been implemented and a more useable and flexible model has been created, there are several areas that could still be considered in future work. These include the introduction of heterogeneous forces, explicit computation of ammunition expenditure and artillery (indirect fire). Possible methods for employing these ideas are discussed below.

A. HETEROGENEOUS FORCES

— The current model involves combat between homogeneous forces only. In other words, each opposing force is comprised entirely of one weapon system type. Added flexibility could be attained by allowing heterogeneous force structures on both sides. This would enable the user to investigate the effect of different force mixes on battle outcome.

The introduction of different weapon systems within a single unit would require extensive changes to the attrition process. Although Lanchester equations for aimed-fire could still be utilized, casualty assesment against each weapon system type by each opposing weapon system type would have to be calculated. The total attrition of any particular unit would then be the sum of the damage assessed to each weapon system of that unit. Additionally, the force level for each weapon system would be required

as an input as well as a separate set of hit and kill probabilities for each weapon system type against each type of target.

Introduction of this feature would create a more complicated model, adding realism but detracting from the current simple and transparent form. Since the purpose of the current effort has been to maintain this simplicity, this option has not been included, but could be considered for future work.

B. AMMUNITION EXPENDITURE

The combat process is a complicated and intricate process. One of the hardest, and not many times attempted, areas to model is the logistic area. However, in the model currently being studied, the problem of ammunition expenditure could be modeled.

Since the model is an expected value type model, it would not be effective to model the amount of ammunition expended by simply counting bullets as they are fired. Actually, in this type of model it would not be possible to count each round, because of the aggregated nature of each unit. The only way to model ammunition useage is to model it in the same way as attrition. This simply means that each unit would have to be given a starting level of ammunition on hand, a basic load, and each time the unit fired, the expected ammount of ammunition expended would be subtracted from what is on hand. The ammount expended would necessarily be a function of the size of the unit

firing, the number of targets fired at and the rate this unit is firing. This can be directly correlated to the attrition-rate coefficient for the firing unit. When ammunition on hand reaches the zero level, the unit would have to be removed from the battle permanently or for some specified time period, to simulate resupply.

C. ARTILLERY

In any armed conflict, there are more ways to inflict casualties than just direct fire. Many types of indirect fire are utilized on the battlefield, including artillery, close air support and naval gunfire. To include this type of play in this model, primarily artillery, would involve a major effort with large program additions.

To model artillery in an expected value model that already uses Lanchester equations for direct fire would suggest the use of Lanchester's area-fire equations. These equations are no more complicated to handle than the aimed-fire equations that are already in use. However, other model considerations would have to be investigated. These would include location of artillery units, whether to use forward observers for target location or only use pre-planned strikes or both and fire allocation procedures to be used. Also, a significant amount of data would need to be collected concerning weapon types to be used, effective ranges, killing radii and accuracy data.

To model artillery would be a significant addition to the model. However, this too would effect the simplicity and transparency the model currently possesses. This addition has also been left for future consideration.

APPENDIX A
User's Manual

I. INTRODUCTION

Smoler's model of land combat is a force-on-force combat model that utilizes Lanchester's aimed fire equations for casualty assessment. The battle is simulated on a 10 x 10 KM piece of terrain representing an area east of Fulda, West Germany. A portion of the terrain map is inclosed as Figure 1. It is a computer model that is coded in FORTRAN and is available for use in the W. R. Church Computer Center.

The purpose of this manual is to familiarize the user with the model and to provide the required Job Control Language (JCL) and user inputs to run the model. A sample listing of program output will also be provided to familiarize the user with the expected program output.

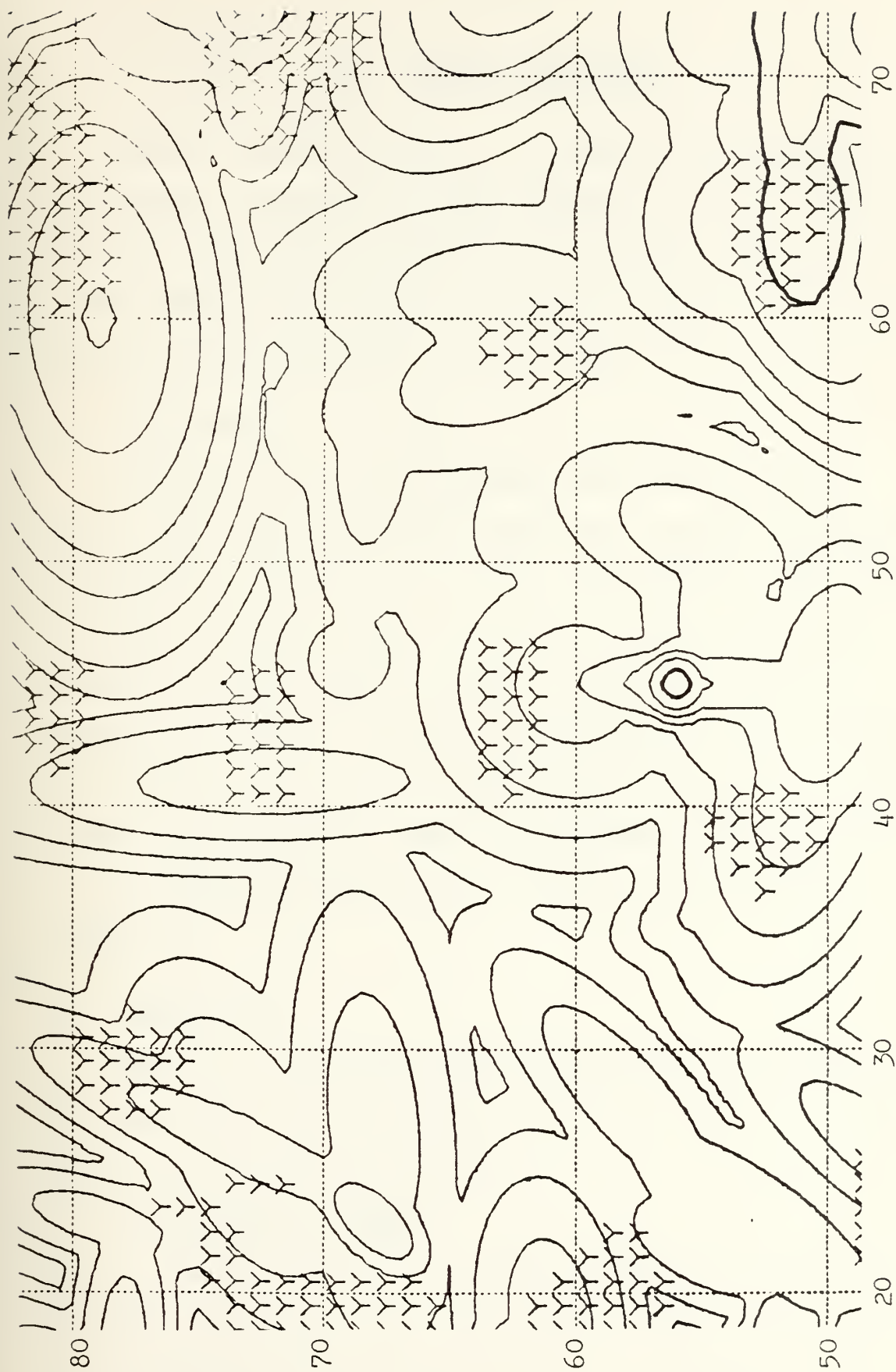


Figure 1. Model Terrain Map

II. PROGRAM STRUCTURE

Smoler's enhanced model is a computerized model, coded in FORTRAN, containing a main program and nine subroutines. To assist the user in understanding how the model operates, a brief description of the function of each of these major parts of the program will be included.

A. MAIN PROGRAM

The main program has several important functions. All of the input and output functions, except the line-of-sight data, are contained in the main program. Additionally, the main program is used to structure all of the other functions during each 10 second time step. Attrition, detection, movement and fire allocation are also handled in the main program. The nine subroutines provide needed input numbers for the above calculations. The general flow of the program is depicted in the flow chart shown at Figure 2.

B. SUBROUTINES

There are nine subroutines included in the model. Each of these subroutines performs a distinct function, each of which will be discussed below.

1. Subroutine SETUP

This subroutine is used to read in the terrain data for the Fulda Map. This terrain data will be used when computing line-of-sight between targets and observers,

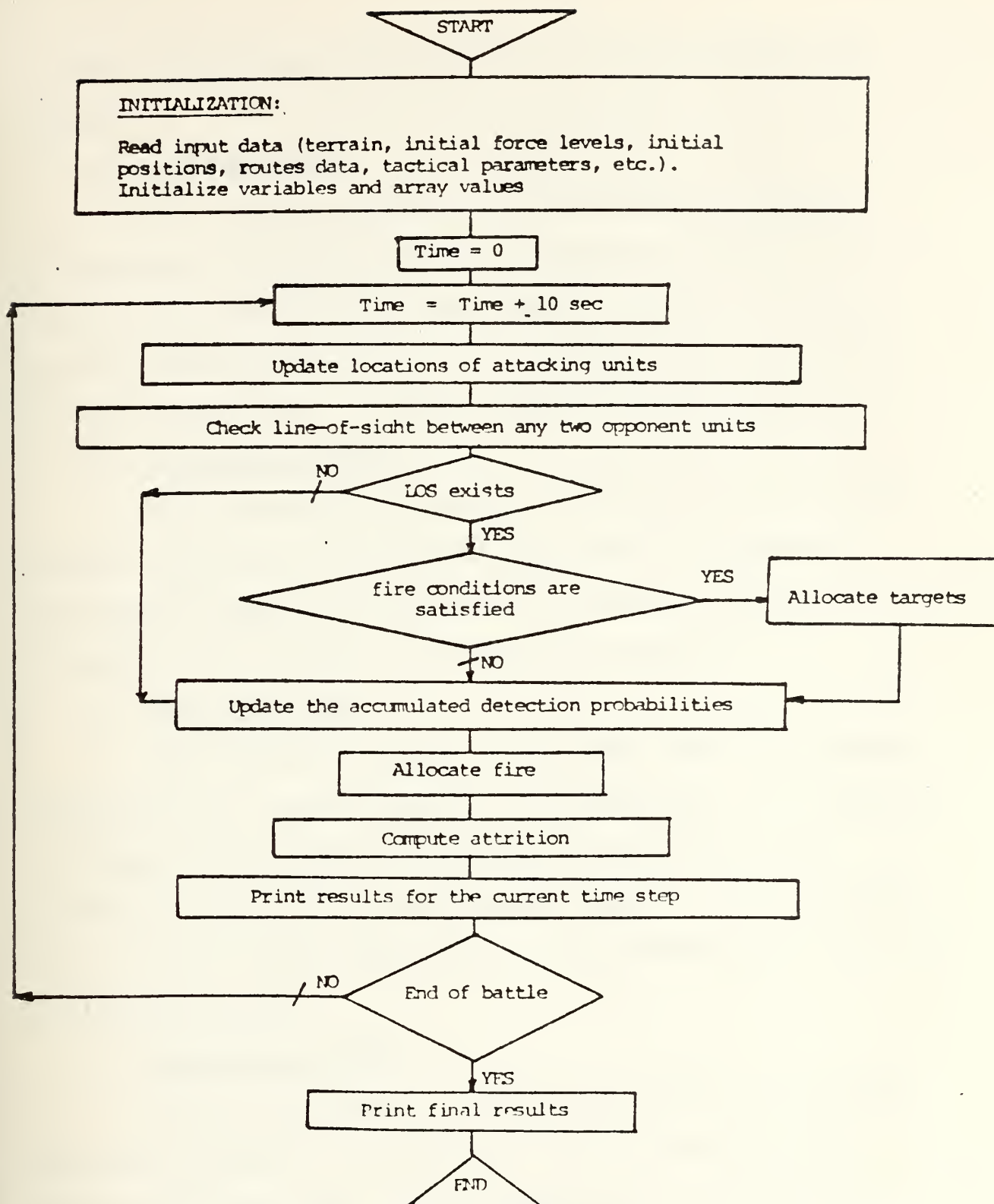


FIGURE 2. FLOW CHART

as well as providing a grid system for unit locations and movement.

2. Subroutine LOS

This module, developed by Professor Jim Hartman of the Naval Postgraduate School, computes a percent of a target visible to a particular observer, given the coordinates (location) of both. This visible fraction is used in the detection and attrition modules in the model.

3. Subroutine KOVER

This subroutine is used by subroutine LOS in determining what portion of a particular target is covered by the terrain between the target and observer. This number is used in the detection and attrition modules.

4. Subroutine ETK

This module computes the expected time for a given firer to kill a given target. The calculation is a function of range, time of flight for a round and hit and kill probabilities for the firing weapon system. It is a number that is used in the computation of the deterministic attrition coefficients, A_{ij} .

5. Subroutine STOCH

This is used when a user has selected a stochastic attrition option to compute the attrition coefficients during each time step. The calculation is a function of the original stochastically determined A_{ij} as well as a function of range.

7. Subroutine LAMBDA

This subroutine is used in conjunction with the LOS routine to compute the detection rate of a target by an observer given the percent of target visible to the observer.

8. Subroutine ROUTE

This is used to compute the route of each attacking red unit when the user has selected the option of inputting attacker routes. It calculates the coordinates of each interval endpoint along the route, making each interval length (distance moved during a 10 second time step) the same. The interval length is determined by the speed the user has selected and inputed for the current battle.

9. Subroutine ELEV

A subroutine that is used to calculate the terrain elevation for a given set of X, Y coordinates. This function is used in conjunction with the LOS subroutine in computing line-of-sight between observer and target.

III. AVAILABLE OPTIONS

The enhanced version of this model has been written to allow for maximum user flexibility while maintaining the simple and transparent nature of the model. To allow this flexibility, there have been several user options incorporated into the model. Each of these options, including user responsibilities, will be discussed here, with the required input data for each being outlined in the next section.

A. STOCHASTIC VS DETERMINISTIC ATTRITION

The user is required to specify whether he wants to use stochastic or deterministic attrition calculations. Both methods utilize Lanchester's aimed-fire equations, the difference being the method of calculating the attrition-rate coefficients. The deterministic procedure uses the Bonder-Farrell method of calculating the A_{ij} , while the stochastic method selects an initial random A_{ij} for each unit, and uses this as a function of range to calculate the A_{ij} for each time step.

B. ATTACK ROUTES

The second major option available to the user is the method of route computation for the attacking forces. The user has to decide whether to allow the program to compute straight west to east routes or to input the routes

he desires each attacking unit to follow. To select his own routes, a user must input the number of nodes he wishes to have on each route and the coordinates of each of these nodes. The program will then compute routes through each node. The nodes must be inputted in order from west to east and should not create an angle between the west to east axis and the route direction that exceeds 45° .

C. ALTERNATE DEFENSIVE POSITIONS

The third option the user must consider is the use of alternative defensive locations. This option permits the user, if he desires, the capability to move the defenders to alternate positions if the attackers close within some specified distance. This breakpoint distance is decided and inputted by the user and is also used as the distance for battle termination. The alternative to moving the defenders is to terminate the battle when the breakpoint distance is reached.

D. OTHER INPUTS

There are other inputs that are required by the program that the user must provide. These include force sizes, weapon characteristics, unit locations and hit and kill probabilities. The required formats for all inputs are outlined in the next section.

IV. INPUT DATA

All user input data is read in from cards at the start of the main program, MAIN. A brief description of each data element as well as the required format for inputting this data is discussed next. A sample data deck has also been included as Figure 3, with the referenced data element number appended on the right. This sample data deck reflects the use of all data elements. However, not all the elements described below are required, and those that are needed only if a particular option is selected are noted with an asteric next to the data element number.

<u>Data Element</u>	<u>Description and Format</u>
1.	Selection of attrition option and random number seed. A 1 is entered for deterministic or a 0 for stochastic attrition. This number is followed by a five digit random number seed. Format for this card is I1,IX,I5.
2.	Number of blue units (NBU) and number of red units (NRU). Both of these elements are two digit integers. Card format is I2,iX,I2.
3.	Effective weapon ranges listed in the following order. Red minimum, red

maximum, blue minimum and blue maximum.

Card format is 4(F6.1,1X).

4. Red force levels for each unit.

Card format is F3.1, 1X. The card will contain the number of entries that equals the number of red units (NRU) in data element 2.

5. Type of route computation desired followed by vehicle speed. Use a 1 for user determined routes or a 0 for program determined routes. A one digit entry will designate desired speed as follows:

1 for 9 mph

2 for 12 mph

3 for 15 mph

4 for 18 mph

Card format for these elements is I2, 1X, I2.

6. X, Y coordinates for each initial red location. One card is needed for each red unit with format F6.1, 1X, F6.1.

7.* If the user has selected to enter his own routes in data element 5, the following is required for each route.

a. The number of nodes in the route (from 01 to 10). This card's format is I2.

b. X, Y coordinates of each node along the route in order from west to east. One card for each node in format F6.1, 1X, F6.1.

8. X, Y coordinates for blue location, force level for each unit, primary search direction for that unit and desired search width. Force level must be between 1.0 and 3.0, search direction between 135° and 225° , and search width from 30° to 120° . One card for each blue unit with format F6.1, 1X, F6.1, 1X, F3.1, 1X, I3, 1X, I3.
9. Specify if blue is to move to alternate defensive positions, breakpoint distance, and number of 10 second time intervals allowed for the move. A 1 for no move or 0 for option to move. Format for this card is I1, 1X, F6.1, 1X, I2.
- 10.* Alternate blue position X and Y coordinates if move is desired (data element 9). One card for each location with format F6.1, 1X, F6.1
11. Red weapon system hit and kill probabilities. The probabilities are entered for 6 range bands, with one card for each range band. Four probabilities for each range are probability of a 1st round hit,

probability of a hit given a previous round hit, probability of a hit given a previous round miss, and probability of a kill given a hit. The 6 range bands are from 0 to 500, 500 to 1000, 1000 to 1500, 1500 to 2000, 2000 to 2500 and 2500 to 3000. Each of these six cards has the format 4(F4.2, 1X).

12. Same as item 11, except the probabilities for the blue weapon systems are entered. Six cards with format 4(F4.2, 1X).

In the formats listed above, a 1X is a space, an Ix is an integer of x digits, and a Fx.y is a real number of length x with y digits to the right of the decimal. For example, an I2 could be a 25, an F6.1 could be 3487.4. A format containing these specifications could be I2, 1X, F6.1. The data card would then look like 25 3487.4, with the 25 beginning in card column one.

card column	1				2				3				4				data element number			
	1	2	8	9	4	3														
	0	3	0	3																
	0	0	0	0	.	0	2	5	0	0	.	0	0	5	0	0	.	0	4	0
	3	.	0	3	.	0	3	.	0											
	1	2																		
	2	5	0	0	.	0	7	3	0	0	.	0								
	2	4	0	0	.	0	5	9	0	0	.	0								
	2	4	5	0	.	0	5	3	0	0	.	0								
	0	2																		
	4	9	0	0	.	0	7	2	0	0	.	0								
	7	1	0	0	.	0	6	7	0	0	.	0								
	0	2																		
	5	3	5	0	.	0	6	0	0	0	.	0								
	7	0	0	0	.	0	6	3	0	0	.	0								
	0	1																		
	6	9	0	0	.	0	5	5	0	0	.	0								
	5	7	0	0	.	0	7	8	0	0	.	0	3	.	0	1	9	0	1	2
	5	8	0	0	.	0	6	5	0	0	.	0	3	.	0	1	8	0	1	2
	5	9	0	0	.	0	5	9	0	0	.	0	3	.	0	1	8	0	1	2
	0	0	5	0	.	0	4													
	7	0	0	0	.	0	7	2	0	0	.	0								
	7	1	0	0	.	0	6	5	0	0	.	0								
	7	1	0	0	.	0	5	8	0	0	.	0								
	0	.	8	5	0	.	0	8	5	0	.	0	7	.	5	0	7	0		
	0	.	8	0	0	.	0	8	0	0	.	0	7	.	5	0	7	0		
	0	.	7	5	0	.	0	7	5	0	.	0	7	.	0	6	6	5		
	0	.	6	0	0	.	0	6	5	0	.	0	6	0	0	5	5			
	0	.	4	5	0	.	0	5	0	0	.	0	5	0	0	3	5			
	0	.	2	0	0	.	0	2	0	0	.	0	2	0	0	2	0			

Figure 3. Sample Data Deck

card column	1										3										4										data element number
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
	0.60	0.70	0.65	0.85																											
	0.85	0.90	0.85	0.90																											
	0.80	0.85	0.85	0.80																											11
	0.75	0.80	0.75	0.70																											
	0.60	0.70	0.65	0.65																											
	0.40	0.45	0.40	0.50																											

Figure 3. Sample Data Deck (cont)

V. REQUIRED JCL

The model and line-of-sight data are residing on permanent disk in the W. R. Church Computer Center. In order to exercise the model, the required job control language (JCL) is illustrated in Figure 4. Due to the CPU time required for execution, a time parameter of TIME = 2 must be used on the job card.

```
//JOB CARD , TIME=2
// EXEC FORTCLG
//FORT.SYSIN DD UNIT=2314,VOL=SER=PAT002,DISP=SHR,
// DSN=S1360.SMOLER.PGM,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
/*
//GO.FTO8FOO1 DD UNIT=2314,VOL=SER=PAT002,DISP=SHR,
// DSN=S1360.LOSDATA,DCB=(RECFM=FB,LRECL=80,BLKSIZE=3200)
/*
//GO.SYSIN DD *
```

USER DATA DECK

/*

Figure 4. Required JCL

VI. EXPECTED OUTPUT

Once the model has been executed, the user can expect the following output:

- 1 - Program listing of the model.
- 2 - A summary of the initial battle conditions, including starting locations and options selected.
- 3 - A summary of battle conditions after each 10 second time step, including unit locations, force levels, unit status, percentage lost and targets on each unit's target list. The status will show one of the following:

- 0 - Unit alive, not firing
- 1 - Unit alive and firing
- 2 - Unit killed
- 3 - Unit moving

- 4 - A statement of the reason for battle termination.

A sample of the initial battle summary is at Figure 5 and the time step summary is at Figure 6.

INITIAL BATTLE INFORMATION

UNIT	X	Y	FORCE LEVEL
1	1900.0	7800.0	6.0
2	1700.0	6500.0	6.0
3	1700.0	4900.0	6.0
4	5800.0	7700.0	3.0
5	7400.0	6300.0	3.0
6	6200.0	5150.0	3.0

ATTRITION IS STOCHASTIC

ROUTES DETERMINED BY USER

RED VEHICLE SPEED IS 15.0

BREAKPOINT DISTANCE IS 1500.0

BLUE WILL MOVE TO ALTERNATE POSITIONS
ALTERNATE POSITIONS ARE:

UNIT	X	Y
4	9300.0	7300.0
5	9800.0	6200.0
6	9200.0	5800.0

RED KILL PROBABILITIES

RANGE	P	PHH	PHM	PKH
500	0.95	0.95	0.95	1.00
1000	0.80	0.83	0.81	0.95
1500	0.65	0.70	0.70	0.90
2000	0.40	0.50	0.55	0.75
2500	0.30	0.40	0.40	0.75
3000	0.20	0.30	0.25	0.60

BLUE KILL PROBABILITIES

RANGE	P	PHH	PHM	PKH
500	0.80	0.80	0.80	1.00
1000	0.80	0.80	0.80	0.95
1500	0.70	0.70	0.70	0.90
2000	0.65	0.65	0.65	0.85
2500	0.45	0.55	0.45	0.70
3000	0.45	0.55	0.45	0.60

Figure 5. Initial Battle Summary

TIME = 480 SECONDS

UNIT	X	Y	FORCE LEVEL	STATUS	LOST-PCT	TARGETS
1	4933.2	7193.2	5.0	0	0.160	
2	4832.3	5799.9	4.7	0	0.214	
3	4902.6	4579.8	6.0	0	0.0	
4	5800.0	7700.0	0.0	2	1.000	
5	7400.0	6300.0	3.0	1	0.0	2
6	6200.0	5150.0	3.0	3	0.0	

Figure 6. Time Step Summary

APPENDIX B

Program Listing

```

1      COMMON /GRP1/ IPRDIR (6), ISECHD (6), MVTDIR (6), X (6), Y (6), SPD (6)
2      COMMON /GRP2/ TA (2), T1 (2), TH (2), TM (2), TF1 (2), TF2 (2), TF3 (2),
3      IP (2, 6), PHH (2, 6), PHM (2, 6), PKH (2, 6), TF (2)
4      COMMON /GRP3/ NBU, NRU, FL (6), FO (6), NOI (3), XIC (3, 200), YIC (3, 200),
5      IIDIR (3, 200), AVSP, ISPD
6      I, IUSTAT (6), II (6), LOST (6, 6), VISFRA, VISFRB, SIZE TK,
7      ISIZETH, NT (6), NF (6), SAF, DISMAX,
8      INLOSC (6, 6), VISFR (6, 6), RMINTK, RMXTK, RMINTW, RMXTW, OP, TOWFR, TNKFR,
9      IPTT (3, 3), RF, POA (6, 6), APOA (6, 6), LOA (6, 6), NA (6), OFL (6), POL (6)
10     COMMON /GRP4/ TPOL (6), OLDQ (6, 6), Q (6, 6)
11     COMMON /GRP5/ LOT (6, 6), ROT (6, 6)
12     COMMON /HILLS/ XC (100), YC (100), PEAK (100), SX (100), SY (100), RHQ (100)
13     COMMON /HILLS/ SCALE (100), TWRHO (100), TWOSCL (100), BASE
14     COMMON /HILLS/ NHILLS
15     COMMON /COVER/ CXC (150), CYC (150), CPEAK (150), CPXX (150), CPTY (150)
16     COMMON /COVER/ CPXY (150), NCVELS
17     COMMON /COUNTR/ KH, KHW, KV, KN, KGAS, KELL, KINT
18     COMMON /GRID/ LST (10, 10), NHL (10, 10), LISTH (450), KHREP (100), KTREP
19     COMMON /GRID/ LSTC (10, 10), NC (10, 10), LISTC (400), KCREP (150)
20     COMMON /GRP6/ ALPHA (6)
21     COMMON /GRP7/ XA (6), YA (6), IMOVE (6)
22     C
23     C      INITIALIZATION.
24     C
25     CALL OVFLOW
26     BL=0.0
27     AL=0.0
28     MP=0
29     PAI=22.0/7.0
30     ZL=.00001
31     CALL SETUP
32     C
33     C      READ TERRAIN DATA FOR LINE OF SIGHT
34     C      CHECK FOR STOCHASTIC OR DETERMINISTIC ATTRITION
35     C
36     READ (5, 130) ITRIT, IS
37     130 FORMAT (I1, 1X, I5)
38     DO 192 I=1, 6
39     CALL RANDOM (IS, YRAN, 1)
40     ALPHA (I) = (-2.*YRAN**2) + (2.*YRAN+.3)
41     192 CONTINUE
42     C
43     C      READ IN NUMBER OF BLUE AND RED UNITS
44     C
45     READ (5, 200) NBU, NRU
46     200 FORMAT (I2, 1X, I2)
47     C
48     C      INITIALIZE WEAPON SIZES
49     C
50     SIZE TK=2.5

```



```

51          SIZETW=2.5
52      C
53      C   READ IN EFFECTIVE WEAPON RANGES
54      C
55          READ (5,102) RMINTK,RMXTK,RMINTW,RMXTW
56      102 FORMAT (F6.1,1X,F6.1,1X,F6.1,1X,F6.1,1X)
57      C
58      C   INITIALIZE PM,RF,TOWFR,TNKFR AND NOD
59      C
60          PM=.352
61          RF=.5
62          TOWFR=.03
63          TNKFR=.1
64          NOD=2
65          DO 101 I=1,NRU
66              NOI(I)=125
67      101 CONTINUE
68          K=NRU+1
69          L=NRU+NBU
70          DO 111 I=1,L
71              II(I)=0
72      111 CONTINUE
73      C
74      C   READ IN FORCE LEVELS OF EACH RED UNIT
75      C
76          READ (5,103) (FL(I),I=1,NRU)
77      103 FORMAT (3 (F3.1,1X))
78      C
79      C   CHECK FOR TYPE OF ROUTE DETERMINATION
80      C
81          READ (5,106) IRTE,ISPD
82      106 FORMAT (I1,1X,I1)
83          IF (ISPD.EQ.1) AVSP=9.0
84          IF (ISPD.EQ.1) DST=40.232
85          IF (ISPD.EQ.2) AVSP=12.0
86          IF (ISPD.EQ.2) DST=53.643
87          IF (ISPD.EQ.3) AVSP=15.0
88          IF (ISPD.EQ.3) DST=67.053
89          IF (ISPD.EQ.4) AVSP=18.0
90          IF (ISPD.EQ.4) DST=80.463
91      C
92      C   READ IN INITIAL RED LOCATIONS
93      C
94          DO 6 I=1,NRU
95              READ (5,107) XIC(I,1),YIC(I,1)
96      107 FORMAT (F6.1,1X,F6.1)
97          6 CONTINUE
98          IF (IRTE.EQ.1) GO TO 108
99          DO 2 I=1,NRU
100          DO 2 J=2,125

```



```

101      YIC(I,J)=YIC(I,J-1)+DST*(J-1)
102      XIC(I,J)=XIC(I,J-1)+DST*(J-1)
103      IDIR(I,J)=0
104      2 CONTINUE
105      GO TO 109
106      108 CALL ROUTE
107      109 SUMR0=0.0
108      DO 3 I=1,NRU
109      FO(I)=FL(I)
110      SUMR0=SUMR0+FO(I)
111      X(I)=XIC(I,1)
112      Y(I)=YIC(I,1)
113      MVTDIR(I)=IDIR(I,1)
114      SPD(I)=AVSP
115      IUSTAT(I)=0
116      IPRDIR(I)=IDIR(I,1)
117      ISECWD(I)=120
118      NF(I)=1
119      II(I)=1
120      3 CONTINUE
121      C
122      C READ IN INITIAL BLUE LOCATIONS
123      C
124      SUMB0=0.0
125      DO 4 I=K,L
126      READ(5,104) X(I),Y(I),FL(I),IPRDIR(I),ISECWD(I)
127      104 FORMAT(F6.1,1X,F6.1,1X,F3.1,1X,I3,1X,I3)
128      FO(I)=FL(I)
129      SUMB0=SUMB0+FO(I)
130      MVTDIR(I)=0
131      SPD(I)=0.0
132      IUSTAT(I)=0
133      IMOVE(I)=0
134      4 CONTINUE
135      C
136      C CHECK FOR ALTERNATE BLUE POSITIONS AND READ IN IF WANTED
137      C
138      READ(5,400) IALT,BREAK,ITEM
139      400 FORMAT(I1,1X,F6.1,1X,I2)
140      IF (IALT.EQ.1) GO TO 401
141      DO 402 I=K,L
142      READ(5,107) XA(I),YA(I)
143      402 CONTINUE
144      401 DELT=10.
145      TA(1)=20.
146      T1(1)=8.
147      TH(1)=8.
148      TM(1)=10.
149      TF1(1)=1.
150      TF2(1)=1.

```



```

151         TF3(1)=1.
152         TA(2)=20.
153         T1(2)=8.
154         TH(2)=8.
155         TM(2)=15.
156         TF1(2)=10.
157         TF2(2)=12.
158         TF3(2)=15.
159     C
160     C   READ IN HIT AND KILL PROBABILITIES
161     C
162         DO 5 I=1,2
163         DO 514 J=1,6
164             READ(5,515) P(I,J),PHH(I,J),PHM(I,J),PKH(I,J)
165     515   FORMAT(4(F4.2,1X))
166     514   CONTINUE
167         5   CONTINUE
168             PTT(1,1)=1.0
169             PTT(1,2)=0.8
170             PTT(2,2)=0.2
171             PTT(1,3)=0.8
172             PTT(2,3)=0.15
173             PTT(3,3)=0.05
174             DO 31 I=1,NRU
175             DO 31 J=K,L
176                 NLOSC(I,J)=0
177                 NLOSC(J,I)=0
178                 Q(I,J)=1.0
179                 Q(J,I)=1.0
180                 VISFR(I,J)=0.0
181                 VISFR(J,I)=0.0
182     31   CONTINUE
183             IC=1
184     C
185     C   PRINT INITIAL BATTLE INFORMATION
186     C
187         WRITE(6,599)
188     599   FORMAT('1',1X,'INITIAL BATTLE INFORMATION')
189         WRITE(6,600)
190     600   FORMAT(/1X,'UNIT',7X,'X',8X,'Y',4X,'FORCE LEVEL')
191         DO 601 I=1,L
192             WRITE(6,602) I,X(I),Y(I),FL(I)
193     602   FORMAT(1X,I3,3X,F7.1,2X,F7.1,7X,F3.1)
194     601   CONTINUE
195             IF(ITRIT.EQ.1) GO TO 603
196             WRITE(6,604)
197     604   FORMAT(/1X,'ATTRITION IS STOCHASTIC'/)
198             GO TO 605
199     603   WRITE(6,606)
200     606   FORMAT(/1X,'ATTRITION IS DETERMINISTIC'/)

```



```

201      605 IF (IRATE.EQ.0) GO TO 607
202      WRITE (6,608)
203      608 FORMAT (1X, 'ROUTES DETERMINED BY USER'//)
204      607 WRITE (6,609) AVSP
205      609 FORMAT (1X, 'RED VEHICLE SPEED IS ',F4.1//)
206      WRITE (6,610) BREAK
207      610 FORMAT (1X, 'BREAKPOINT DISTANCE IS ',F6.1//)
208      IF (IALT.EQ.0) GO TO 615
209      WRITE (6,620)
210      620 FORMAT (1X, 'BLUE WILL NOT MOVE TO ALTERNATE POSITIONS'//)
211      GO TO 625
212      615 WRITE (6,630)
213      630 FORMAT (1X, 'BLUE WILL MOVE TO ALTERNATE POSITIONS'//1X,
214      1 'ALTERNATE POSITIONS ARE:'//1X, 'UNIT',5X, 'X',8X, 'Y')
215      DO 635 I=K,L
216      WRITE (6,640) I,XA(I),YA(I)
217      640 FORMAT (1X,I3,3X,F7.1,2X,F7.1)
218      635 CONTINUE
219      625 IRAN=500
220      WRITE (6,645)
221      645 FORMAT (/4X, 'RED KILL PROBABILITIES'//1X, 'RANGE',4X, 'P',
222      14X, 'PHH',3X, 'PHM',3X, 'PKH')
223      DO 650 I=1,6
224      WRITE (6,655) IRAN,P (1,I),PHH (1,I),PHM (1,I),PKH (1,I)
225      655 FORMAT (2X,I4,4 (2X,F4.2))
226      IRAN=IRAN+500
227      650 CONTINUE
228      IRAN=500
229      WRITE (6,660)
230      660 FORMAT (/4X, 'BLUE KILL PROBABILITIES'//1X, 'RANGE',4X, 'P',
231      14X, 'PHH',3X, 'PHM',3X, 'PKH')
232      DO 665 I=1,6
233      WRITE (6,655) IRAN,P (2,I),PHH (2,I),PHM (2,I),PKH (2,I)
234      IRAN=IRAN+500
235      665 CONTINUE
236      WRITE (6,670)
237      670 FORMAT ('I',10X, 'BATTLE BEGINS'//)
238      C
239      C      UPDATE LOCATION OF RED UNITS.
240      C
241      DISMAX=5000.0
242      67 DO 9 I=1,NAU
243      IF (IUSTAT(I).EQ.2) GOTO 9
244      IF (IUSTAT(I).EQ.0) GOTO 76
245      NF(I)=NF(I)+1
246      IF (NF(I).LT.N00) GOTO 9
247      NF(I)=1
248      76 DO 11 J = 1, NAU
249      IF (J.EQ. I) GO TO 11
250      IF (IUSTAT(J).EQ. 2) GO TO 11

```



```

251      DIST = X(I) - X(J)
252      IF (DIST .GT. DISMAX) GO TO 9
253  11 CONTINUE
254      II(I) = II(I) + 1
255      K7=II(I)
256      X(I)=XIC(I,K7)
257      Y(I)=YIC(I,K7)
258      MVTDIR(I)=IDIR(I,K7)
259      IPDIR(I)=IDIR(I,K7)
260  9 CONTINUE
261  C
262  C      LINE--OF-SIGHT CHECK BETWEEN UNITS AND TARGETS SELECTION
263  C
264      DO 17 J=K,L
265      NT(J)=0
266  17 CONTINUE
267      DO 12 I=1,NRU
268      NT(I)=0
269      IF (IUSTAT(I).EQ.2) GOTO 12
270      DO 16 J=K,L
271      IF (IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 16
272      XX1=X(I)
273      YY1=Y(I)
274  133 CALL ELEV(XX1,YY1,TMACI)
275      XX2=X(J)
276      YY2=Y(J)
277      CALL ELEV(XX2,YY2,TMACJ)
278      LATOB=1
279      LBTOA=1
280      CALL LOS(XX1,YY1,TMACI,0.0,SIZETK,XX2,YY2,TMACJ,0.0,SIZETH,
281  1LATOB,LBTOA,VISFRA,VISFRB)
282      VISFR(I,J)=VISFRA
283      VISFR(J,I)=VISFRB
284      IF (VISFRA.GT.ZL) GOTO 18
285      LOST(I,J)=0
286      LOST(J,I)=0
287      NLOSC(I,J)=NLOSC(I,J)+1
288      NLOSC(J,I)=NLOSC(I,J)
289      GOTO 16
290  18 LOST(I,J)=1
291      LOST(J,I)=1
292      NLOSC(I,J)=0
293      NLOSC(J,I)=0
294      RANGE=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
295      IF (RANGE.LT.RMINTK.OR.RANGE.GT.RMXTK) GOTO 20
296      IF (Q(I,J).EQ.1.0) GOTO 20
297      IUSTAT(I)=1
298      NT(I)=NT(I)+1
299      M=NT(I)
300      LOT(I,M)=J

```



```

301      ROT(I,M)=RANGE
302      IF (M.EQ.1) GOTO 20
303      CALL SORT(I,M)
304  20 IF (RANGE.LT.RMINTW.OR.RANGE.GT.RMXTW) GOTO 16
305      IF (Q(J,I).EQ.1.0) GOTO 16
306      IUSTAT(J)=1
307      NT(J)=NT(J)+1
308      M=NT(J)
309      LOT(J,M)=I
310      ROT(J,M)=RANGE
311      IF (M.EQ.1) GOTO 16
312      CALL SORT(J,M)
313  16 CONTINUE
314  12 CONTINUE
315      DO 25 I=1,NRU
316      IF (IUSTAT(I).EQ.2) GOTO 25
317      IF (NT(I).NE.0) GOTO 25
318      IUSTAT(I)=0
319      NF(I)=0
320  25 CONTINUE
321      DO 79 J=K,L
322      IF (IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 79
323      IF (NT(J).EQ.0) IUSTAT(J)=0
324  79 CONTINUE
325  C
326  C      UPDATE OF THE ACCUMULATED DETECTION PROBABILITIES.
327  C
328      IAA=1
329      IBB=NRU
330      ICC=K
331      IDD=L
332      FR=TOWFR
333      OP=PM
334  37 DO 14 I=IAA,IBB
335      IF (IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 14
336      DO 19 J=ICC,IDD
337      PROP=0.0
338      IF (IUSTAT(J).EQ.2.OR.IUSTAT(J).EQ.3) GO TO 19
339      OLDQ(I,J)=Q(I,J)
340      IF (LOST(I,J).EQ.0) GOTO 15
341      IF (NT(I).GT.0) GOTO 22
342      PCTVIS=VISFR(J,I)
343      CALL LAMDA(I,J,PCTVIS,DETRAT,PSUBK)
344      QV=EXP(-(FL(I)*DETRAT*OP*DELT*FL(J)))
345      IF (NT(J).GT.0) GOTO 23
346      Q(I,J)=Q(I,J)*QV
347      GOTO 19
348  23 QP=(1.0-PSUBK)**(FR*DELT*FL(I)*FL(J))
349      Q(I,J)=Q(I,J)*(QV+QP-QV*QP)
350      GOTO 19

```



```

351      22 N5=NT(I)
352      DO 24 I1=1,N5
353      K1=LOT(I,I1)
354      ANG1=ATAN2(Y(K1)-Y(I),X(K1)-X(I))
355      ANG2=ATAN2(Y(J)-Y(I),X(J)-X(I))
356      IF((ANG1*ANG2).GE.0.0) GOTO 77
357      IF(ANG2.LT.0.0) GOTO 32
358      ANG=2*PAI+ANG1-ANG2
359      GOTO 35
360      32 ANG=2*PAI+ANG2-ANG1
361      35 IF(ANG.GT.PAI) ANG=2*PAI-ANG
362      GOTO 33
363      77 ANG=ABS(ANG2-ANG1)
364      33 AA=15.0*PAI/180.0
365      IF(ANG.GT.AA) GOTO 24
366      PROP=PROP+PTT(I1,N5)
367      24 CONTINUE
368      IF(PROP.EQ.0.0) GOTO 34
369      IF(NT(J).GT.0) GOTO 36
370      CALL LAMDA(I,J,PCTVIS,DETRAT,PSUBK)
371      DETRAT=DETRAT*RF
372      QV=EXP(-(FL(I)*PROP*DETRAT*DELT*FL(J)))
373      Q(I,J)=Q(I,J)*QV
374      GOTO 19
375      36 Q(I,J)=0.0
376      GOTO 19
377      34 IF(IAA.EQ.1) GOTO 19
378      Q(I,J)=1.0
379      GOTO 19
380      15 IF(NLOSC(I,J).LE.3) GOTO 19
381      Q(I,J)=1.0
382      19 CONTINUE
383      14 CONTINUE
384      IF(IAA.EQ.K) GOTO 38
385      FR=TNKFR
386      IAA=K
387      IBB=L
388      ICC=1
389      IDD=NRU
390      OP=1.0
391      GOTO 37
392      C
393      C      FIRE ALLOCATION.
394      C
395      38 DO 28 I=1,L
396      28 NA(I)=0
397      DO 26 I=1,L
398      IF(IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 26
399      IF(NT(I).EQ.0) GOTO 26
400      DO 27 J=1,3

```



```

401      APOA(I,J)=0.0
402  27 CONTINUE
403      IF (NT(I).EQ.1) GOTO 78
404      IF (NT(I).EQ.2) GOTO 29
405      NOT=3
406      MM1=LOT(I,1)
407      MM2=LOT(I,2)
408      MM3=LOT(I,3)
409      PROB=(1.0-Q(I,MM1))*Q(I,MM2)*Q(I,MM3)
410      APOA(I,1)=APOA(I,1)+PTT(1,1)*PROB
411      PROB=Q(I,MM1)*(1.0-Q(I,MM2))*Q(I,MM3)
412      APOA(I,2)=APOA(I,2)+PTT(1,1)*PROB
413      PROB=Q(I,MM1)*Q(I,MM2)*(1.0-Q(I,MM3))
414      APOA(I,3)=APOA(I,3)+PTT(1,1)*PROB
415      PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))*Q(I,MM3)
416      APOA(I,1)=APOA(I,1)+PTT(1,2)*PROB
417      APOA(I,2)=APOA(I,2)+PTT(2,2)*PROB
418      PROB=(1.0-Q(I,MM1))*Q(I,MM2)*(1.0-Q(I,MM3))
419      APOA(I,1)=APOA(I,1)+PTT(1,2)*PROB
420      APOA(I,3)=APOA(I,3)+PTT(2,2)*PROB
421      PROB=Q(I,MM1)*(1.0-Q(I,MM2))*(1.0-Q(I,MM3))
422      APOA(I,2)=APOA(I,2)+PTT(1,2)*PROB
423      APOA(I,3)=APOA(I,3)+PTT(2,2)*PROB
424      PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))*(1.0-Q(I,MM3))
425      APOA(I,1)=APOA(I,1)+PTT(1,3)*PROB
426      APOA(I,2)=APOA(I,2)+PTT(2,3)*PROB
427      APOA(I,3)=APOA(I,3)+PTT(3,3)*PROB
428  30 DO 44 J=1,NOT
429      KK=LOT(I,J)
430      NA(KK)=NA(KK)+1
431      IN=NA(KK)
432      LOA(KK,IN)=1
433      POA(KK,IN)=APOA(I,J)
434  44 CONTINUE
435      GOTO 26
436  29 NOT=2
437      MM1=LOT(I,1)
438      MM2=LOT(I,2)
439      PROB=(1.0-Q(I,MM1))*Q(I,MM2)
440      APOA(I,1)=APOA(I,1)+PTT(1,1)*PROB
441      PROB=Q(I,MM1)*(1.0-Q(I,MM2))
442      APOA(I,2)=APOA(I,2)+PTT(1,1)*PROB
443      PROB=(1.0-Q(I,MM1))*(1.0-Q(I,MM2))
444      APOA(I,1)=APOA(I,1)+PTT(1,2)*PROB
445      APOA(I,2)=APOA(I,2)+PTT(2,2)*PROB
446      GOTO 30
447  78 NOT=1
448      MM1=LOT(I,1)
449      PROB=1.0-Q(I,MM1)
450      APOA(I,1)=APOA(I,1)+PTT(1,1)*PROB

```



```

451      GOTO 30
452      26 CONTINUE
453      C
454      C      ATTRITION.
455      C
456          SUMR=0.0
457          SUMB=0.0
458          DO 40 I=1,L
459              IF (IUSTAT(I).EQ.2.OR.IUSTAT(I).EQ.3) GO TO 40
460              M6=NA(I)
461              SUM=0.0
462              IF (M6.EQ.0) GOTO 47
463              DO 41 J=1,M6
464                  M7=L0A(I,J)
465                  IF (M7.LT.K) GOTO 42
466                  ITYPE=2
467                  GOTO 43
468              42 ITYPE=1
469              43 RANGE=SQRT((X(I)-X(M7))**2+(Y(I)-Y(M7))**2)
470                  IF (ITRIT.EQ.1) GO TO 131
471                  CALL STOCH(ITYPE,RANGE,AJI)
472                  GO TO 5000
473          131 CALL ETK(ITYPE,RANGE,T)
474              AJI=1.0/T
475          5000      SUM=SUM+AJI*FL(M7)*P0A(I,J)*DELT
476              41 CONTINUE
477              47 OFL(I)=FL(I)
478                  FL(I)=FL(I)-SUM
479                  IF (FL(I).GT.ZL) GOTO 46
480                  FL(I)=0.0
481                  IUSTAT(I)=2
482              46 IF (I.LT.K) GOTO 60
483                  SUMB=SUMB+FL(I)
484                  TPOL(I)=(FO(I)-FL(I))/FO(I)
485                  GO TO 40
486          60 SUMR=SUMR+FL(I)
487                  TPOL(I)=(FO(I)-FL(I))/FO(I)
488              40 CONTINUE
489      C
490      C      PRINT AND CHECK FOR BATTLE DETERMINATION.
491      C
492          ITIME=IC*10
493          DO 57 I=K,L
494              IF (IUSTAT(I).EQ.2) GO TO 57
495              DO 58 J=1,NRU
496                  IF (IUSTAT(J).EQ.2) GO TO 58
497                  CHECK=X(I)-X(J)
498                  AVD=SQRT((X(I)-X(J))**2+(Y(I)-Y(J))**2)
499                  IF (AVD.LT.BREAK.OR.CHECK.LT.50.) GO TO 250
500          58 CONTINUE

```



```

501      57 CONTINUE
502      GO TO 99
503      C
504      C   COMPLETE BLUE MOVE
505      C
506      250 DO 251 I=K,L
507          IF (IALT.EQ.1.OR.IMOVE(I).EQ.ITEM) GO TO 6000
508          IF (IUSTAT(I).EQ.0) IUSTAT(I)=3
509          IMOVE(I)=IMOVE(I)+1
510          IF (IMOVE(I).LT.ITEM) GO TO 251
511          X(I)=XA(I)
512          Y(I)=YA(I)
513          IF (IUSTAT(I).EQ.3) IUSTAT(I)=0
514      251 CONTINUE
515      99 WRITE(6,112) ITIME
516      112 FORMAT(////1X,'TIME = ',14,1X,'SECONDS'//)
517      WRITE(6,113)
518      113 FORMAT(1X,'UNIT',5X,'X',8X,'Y',5X,'FORCE LEVEL',2X,'STATUS',
519      12X,'LOST-PCT',2X,'TARGETS')
520      DO 59 I=1,L
521          N6=NT(I)
522          IF (N6.NE.0) GO TO 48
523          WRITE(6,264) I,X(I),Y(I),FL(I),IUSTAT(I),TPOL(I)
524      264 FORMAT(3X,11,3X,F7.1,2X,F7.1,6X,F3.1,9X,11,6X,F5.3)
525          GO TO 59
526      48 WRITE(6,114) I,X(I),Y(I),FL(I),IUSTAT(I),TPOL(I),
527      1 (LOT(I,J),J=1,N6)
528      114 FORMAT(3X,11,3X,F7.1,2X,F7.1,6X,F3.1,9X,11,6X,F5.3,3X,3(11,1X))
529      59 CONTINUE
530      C
531      C   CHECK FOR BATTLE DETERMINATION.
532      C
533          IOT=0
534          DO 53 I=1,NRU
535              IF (FL(I).EQ.0.0) GOTO 53
536              IOT=1
537      53 CONTINUE
538          IF (IOT.EQ.1) GOTO 54
539          WRITE(6,117)
540      117 FORMAT(1X,'*** RED FORCE IS ELIMINATED.  END OF BATTLE.')
541          GOTO 66
542      54 IOT=0
543          DO 55 I=K,L
544              IF (FL(I).EQ.0.0) GOTO 55
545              IOT=1
546      55 CONTINUE
547          IF (IOT.EQ.1) GOTO 65
548          WRITE(6,118)
549      118 FORMAT(1X,'*** BLUE FORCE IS ELIMINATED.  END OF BATTLE.')
550          GOTO 66

```



```
551      6000 WRITE (6,119)
552      119 FORMAT (1X,'*** DISTANCE BETWEEN FORCES IS TOO CLOSE. END OF BATTLE
553          1')
554          GOTO 66
555      65 IC=IC+1
556          GOTO 67
557      66 STOP
558          END
```



```

1      SUBROUTINE LOS(XA,YA,TMACA, TMICA, SIZEA, XB,YB,TMACB, TMICB, SIZEB,
2      -LATOB, LBTOB, VISFRA, VISFRB)
3
4      C
5      COMMON /HILLS/ XC(100),YC(100),PEAK(100),SX(100),SY(100),RHO(100)
6      COMMON /HILLS/ SCALE(100),TWO RHO(100),TWO SCL(100),BASE
7      COMMON /HILLS/ NHILLS
8      COMMON /COVER/ CXC(150),CYC(150),CPEAK(150),CPXX(150),CPYY(150)
9      COMMON /COVER/ CPXY(150),NCVELS
10     COMMON /COUNTR/KH, KHW, KV, KN, KGRS, KELL, KINT
11     COMMON /GRID/ LST(10,10),NHL(10,10),LISTH(450),KHREP(100),KTREP
12     COMMON /GRID/ LSTC(10,10),NC(10,10),LISTC(400),KCREP(150)
13     DIMENSION IGX(100),IGY(100),IEL(100),CS1(100),CS2(100)
14     DATA NGRID/10/,Gsize/1000./
15
16     C SUBROUTINE TO COMPUTE FRACTION VISIBLE FOR OBSERVER TARGET PAIRS
17     VISFRA=1.
18     VISFRB=1.
19     XBA=XB-XA
20     YBA=YB-YA
21     IF((XBA.EQ.0.).AND.(YBA.EQ.0.)) RETURN
22     IF(SIZEA+TMICA.LE.0.) GO TO 510
23     IF(SIZEB+TMICB.LE.0.) GO TO 510
24     IF(TMICA.LT.0.) VISFRA=1.0+TMICA/SIZEA
25     IF(TMICB.LT.0.) VISFRB=1.0+TMICB/SIZEB
26     ZA=TMACA + TMICA + SIZEA
27     ZB=TMACB + TMICB + SIZEB
28     KTREP=KTREP+1
29     ZBA=ZB-ZA
30     XBASQ=XBA*XBA
31     YBASQ=YBA*YBA
32     XYBA=XBA*YBA
33     TWOXBA=2.*XBA
34     TWOYBA=2.*YBA
35
36     C COMPUTE GRID SQUARES CROSSED BY A TO B LINE
37     NGRSQ=0
38     IF(XBA) 110,95,100
39
40     95     XBA=0.1
41     100    ISGX=-1
42     XINC=Gsize/XBA
43     GO TO 120
44
45     110    ISGX=1
46     XINC=-Gsize/XBA
47     IF(YBA) 140,125,130
48
49     125    YBA=0.1
50     130    ISGY=-1
51     YINC=Gsize/YBA
52     GO TO 150
53
54     140    ISGY=1
55     YINC=-Gsize/YBA
56     150    IX=1+IFIX(XB/Gsize)
57     IF(IX.GT.NGRID) IX=NGRID

```



```

51      IY=1+IFIX(YB/GSIZE)
52      IF(IY.GT.NGRID) IY=NGRID
53      XNEXT=GSIZE*(FLOAT(IX)+0.5*(ISGX-1.))
54      YNEXT=GSIZE*(FLOAT(IY)+0.5*(ISGY-1.))
55      XSTEP=(XB-XNEXT)/XBA
56      YSTEP=(YB-YNEXT)/YBA
57      160  NGRSQ=NGRSQ+1
58          IGX(NGRSQ)=IX
59          IGY(NGRSQ)=IY
60          IF((XSTEP.GT.1.).AND.(YSTEP.GT.1.)) GO TO 200
61          IF(XSTEP-YSTEP) 170,180,190
62      170  IX=IX+ISGX
63          XSTEP=XSTEP+XINC
64          GO TO 160
65      180  IX=IX+ISGX
66          XSTEP=XSTEP+XINC
67      190  IY=IY+ISGY
68          YSTEP=YSTEP+YINC
69          GO TO 160
70      200  KGRS=KGRS+NGRSQ
71      C GRID SQUARE LIST NOW COMPLETE IN IGX, IGY WITH NGRSQ ENTRIES
72      C
73      C NOW FIND WHICH COVER ELLIPSES TOUCH THE A TO B LINE,
74      C CHECK ELEVATIONS AT S1 AND S2 FOR EACH SUCH ELLIPSE
75          NELS=0
76          CHTMAX=0.
77          IF(NCVELS.EQ.0) GOTO 270
78          DO 260 K=1,NGRSQ
79              IX=IGX(K)
80              IY=IGY(K)
81              N=NC(IX,IY)
82              IF(N.EQ.0) GO TO 260
83              LS=LSTC(IX,IY)
84              LEND=LS+N-1
85              DO 250 L=LS,LEND
86                  KELL=KELL+1
87                  IC=LISTC(L)
88                  IF(KCREP(IC).EQ.KTREP) GO TO 250
89                  KCREP(IC)=KTREP
90                  RX=XA-CXC(IC)
91                  RY=YA-CYC(IC)
92                  PXX=CPXX(IC)
93                  PYY=CPYY(IC)
94                  PXY=CPXY(IC)
95                  AA=PXX*XBASQ+PYY*YBASQ+PXY*XYBA
96                  BB=PXX*TWOXBA*RX+PYY*TWOYBA*RY+PXY*(RX*YBA+RY*XBA)
97                  CC=PXX*RX*RX+PYY*RY*RY+PXY*RX*RY-1.0
98                  ARG=BB*BB-4.0*AA*CC
99                  IF(ARG.LE.0.) GO TO 250
100                 SQ=SQRT(ARG)

```



```

101      S1 = - (BB + SQ) / (2.0 * AA)
102      S2 = (SQ - BB) / (2.0 * AA)
103      IF (S1.GE.1.) GO TO 250
104      IF (S2.LE.0.) GO TO 250
105      IF (S1.LE.0.) GO TO 510
106      IF (S2.GE.1.) GO TO 510
107  C NOW CHECK LOS AT S1 AND S2
108      KINT = KINT + 1
109      CPK = CPEAK (IC)
110      XS = XA + S2 * XBA
111      YS = YA + S2 * YBA
112      CALL ELEV (XS, YS, HTS)
113      HTS = HTS + CPK
114      ZS = ZA + S2 * ZBA
115      IF (LATOB.EQ.0) GO TO 210
116      CALL KOVER (ZA, TMACB, SIZEB, ZB, S2, HTS, ZS, VISFAB)
117      IF (VISFAB.LE.0.) GO TO 510
118  210    IF (LBTOA.EQ.0) GO TO 220
119      S = 1. - S2
120      CALL KOVER (ZB, TMACA, SIZEA, ZA, S, HTS, ZS, VISFRA)
121      IF (VISFRA.LE.0.) GO TO 510
122  220    XS = XA + S1 * XBA
123      YS = YA + S1 * YBA
124      CALL ELEV (XS, YS, HTS)
125      HTS = HTS + CPK
126      ZS = ZA + S1 * ZBA
127      IF (LATOB.EQ.0) GO TO 230
128      CALL KOVER (ZA, TMACB, SIZEB, ZB, S1, HTS, ZS, VISFAB)
129      IF (VISFAB.LE.0.) GO TO 510
130  230    IF (LBTOA.EQ.0) GO TO 240
131      S = 1.0 - S1
132      CALL KOVER (ZB, TMACA, SIZEA, ZA, S, HTS, ZS, VISFRA)
133      IF (VISFRA.LE.0.) GO TO 510
134  240    NELS = NELS + 1
135      IEL (NELS) = IC
136      CS1 (NELS) = S1
137      CS2 (NELS) = S2
138      IF (CPK.GT.CHTMAX) CHTMAX = CPK
139  250    CONTINUE
140  260    CONTINUE
141  C ALL ELLIPSES CHECKED
142  C
143  C NOW START ON THE HILLS
144  270 DO 600 K=1,NGRSQ
145      IX = IGX (K)
146      IY = IGY (K)
147      IF (NHL (IX, IY).EQ.0) GO TO 600
148      LS = LST (IX, IY)
149      LEND = LS + NHL (IX, IY) - 1
150      DO 500 L=LS,LEND

```



```

151      I=LISTH(L)
152      IF (KHREP(I).EQ.KTREP) GO TO 500
153      KHREP(I)=KTREP
154  C   PROCESSING FOR HILL I STARTS HERE
155      KH=KH+1
156  C   COMPUTE W =TOP OF THIS HILL ALONG O-T LINE
157      CX=XBA/SX(I)
158      CY=YBA/SY(I)
159      DX=(XA-XC(I))/SX(I)
160      DY=(YA-YC(I))/SY(I)
161      FQ=TWOSCL(I)*(CX*DX+CY*DY+RHO(I)*(CX*OY+CY*DX))
162      GQ=SCALE(I)*(CX*CX+CY*CY+TWO RHO(I)*(CX*CY)
163      IF (GQ.EQ.0.) GO TO 500
164      W=-FQ/(2.*GQ)
165      IF (ABS(W).GT.5.) GO TO 500
166      FSQ=FQ*FQ
167      EQ=SCALE(I)*(DX*DX+DY*DY+TWO RHO(I)*DX*DY)
168      POWER=EQ-FSQ/(4.*GQ)
169      IF (POWER.LT.-3.) GO TO 500
170      HHW=PEAK(I)*EXP(POWER)
171      KHW=KHW+1
172      IF (HHW.LE.BASE) GO TO 500
173      ZW=ZA+W*ZBA
174      IF ((W.LT.0.).OR.(W.GT.1.)) GO TO 300
175      IF (HHW.GE.ZW) GO TO 510
176      CVHTW=0.
177      IF (NELS.EQ.0) GO TO 300
178      DO 280 M=1,NELS
179      IF ((CS1(M).GE.W).OR.(CS2(M).LE.W)) GO TO 280
180      IC=IEL(M)
181      IF (CVHTW.LT.CPEAK(IC)) CVHTW=CPEAK(IC)
182 280   CONTINUE
183      IF ((HHW+CVHTW).GE.ZW) GO TO 510
184 300   IF (HHW+CHTMAX.LT.AMIN1(ZA-SIZEA,ZB-SIZEB)) GO TO 500
185  C   IF WE GET TO HERE THEN NEED TO FIND LOWEST SIGHT LINE OVER HILL
186  C   NEWTON ITERATION A TO B GIVING VISFRB
187      IF (LATOB.EQ.0) GO TO 400
188      KV=KV+1
189      V=W
190      HHV=HHW
191      NCT=0
192      FV=FQ*V
193      TWOGV=2.*GQ*V
194 330   FCNV=ZA+HHV*(TWOGV*V+FV-1.)
195      KN=KN+1
196      FACTOR=(TWOGV*TWOGV+2.*(GQ+TWOGV*FQ)+FSQ)
197      DFCNV=HHV*V*FACTOR
198      IF (ABS(DFCNV).LT.1.E-10) GO TO 350
199      V=V-FCNV/DFCNV
200      IF (ABS(V).GT.5.) GO TO 400

```



```

201      FV=FQ×V
202      TWOGV=2.×GQ×V
203      POWER = EQ+FV+GQ×V×V
204      IF (POWER .LT. -3.) GO TO 400
205      HHV=PEAK (I)×EXP (POWER)
206      DHHV=HHV×(FQ+TWOGV)
207      ELV=ZA+DHHV×V
208      IF (ABS (HHV-ELV) .LT.1.) GO TO 350
209      NCT=NCT+1
210      IF (NCT.LT.10) GO TO 330
211 350   IF ((V.LT.0.) .OR. (V.GT.1.)) GO TO 400
212      CVHTV=0.
213      IF (NELS.EQ.0) GO TO 390
214      DO 380 M=1,NELS
215      IF ((CS1 (M) .GE.V) .OR. (CS2 (M) .LE.V)) GO TO 380
216      IC=IEL (M)
217      IF (CVHTV.LT.CPEAK (IC)) CVHTV=CPEAK (IC)
218 380   CONTINUE
219 390   HTV=HHV+CVHTV
220      ZV=ZA+V×ZBA
221      CALL KOVER (ZA, TMACB, SIZEB, ZB, V, HTV, ZV, VISFRB)
222      IF (VISFRB.LE.0.) GO TO 510
223  C NEWTON ITERATION B TO A GIVING VISFRA
224 400   IF (LBTOA.EQ.0) GO TO 500
225      KV=KV+1
226      V=W
227      VM1=V-1.
228      HHV=HHW
229      NCT=0
230      FV=FQ×V
231      TWOGV=2.×GQ×V
232 430   FCNV=ZB+HHV×((FQ+TWOGV)×VM1-1.)
233      KN=KN+1
234      FACTOR=(TWOGV×TWOGV+2.×(GQ+TWOGV×FQ)+FSQ)
235      DFCNV=HHV×VM1×FACTOR
236      IF (ABS (DFCNV) .LT.1.E-10) GO TO 450
237      V=V-FCNV/DFCNV
238      IF (ABS (V) .GT.5.) GO TO 500
239      VM1=V-1.
240      FV=FQ×V
241      TWOGV=2.×GQ×V
242      POWER = EQ+FV+GQ×V×V
243      IF (POWER .LT. -3.) GO TO 500
244      HHV=PEAK (I)×EXP (POWER)
245      DHHV=HHV×(FQ+TWOGV)
246      ELV=ZB+DHHV×VM1
247      IF (ABS (HHV-ELV) .LT.1.) GO TO 450
248      NCT=NCT+1
249      IF (NCT.LT.10) GO TO 430
250 450   IF ((V.LT.0.) .OR. (V.GT.1.)) GO TO 500

```



```

251      CVHTV=0.
252      IF (NELS.EQ.0) GO TO 490
253      DO 480 M=1,NELS
254      IF ((CS1(M).GE.V).OR.(CS2(M).LE.V)) GO TO 480
255      IC=IEL(M)
256      IF (CVHTV.LT.CPEAK(IC)) CVHTV=CPEAK(IC)
257 480    CONTINUE
258 490    HTV=HHV+CVHTV
259      ZV=ZA+V*ZBA
260      S=-VM1
261      CALL KOVER(ZB,TMACA,SIZEA,ZA,S,HTV,ZV,VISFRA)
262      IF (VISFRA.LE.0.) GO TO 510
263 500    CONTINUE
264 600    CONTINUE
265      RETURN
266 510    VISFRA=0.
267      VISFRB=0.
268      RETURN
269      END

```



```

1      SUBROUTINE SETUP
2      C
3      COMMON /HILLS/ XC (100),YC (100),PEAK (100),SX (100),SY (100),RHO (100)
4      COMMON /HILLS/ SCALE (100),TWO RHO (100),TWO SCL (100),BASE
5      COMMON /HILLS/ NHILLS
6      COMMON /COVER/ CXC (150),CYC (150),CPEAK (150),CPXX (150),CPYY (150)
7      COMMON /COVER/ CPXY (150),NCVELS
8      COMMON /COUNTR/KH,KHW,KV,KN,KGRS,KELL,KINT
9      COMMON /GRID/ LST (10,10),NHL (10,10),LISTH (450),KHREP (100),KTREP
10     COMMON /GRID/ LSTC (10,10),NC (10,10),LISTC (400),KCREP (150)
11     READ (8,7) NHILLS
12     READ (8,47) BASE
13     47  FORMAT (F10.4)
14     7   FORMAT (I6)
15     17  FORMAT (5F8.3,F6.4)
16     DO 50 I=1,NHILLS
17     READ (8,17) XC (I),YC (I),PEAK (I),SX (I),SY (I),RHO (I)
18     50  CONTINUE
19     READ (8,37) LST
20     READ (8,37) NHL
21     READ (8,7) NHTOT
22     READ (8,37) (LISTH (I),I=1,NHTOT)
23     37  FORMAT (16I5)
24     READ (8,7) NCVELS
25     IF (NCVELS.EQ.0) GO TO 65
26     DO 60 I=1,NCVELS
27     READ (8,27) CXC (I),CYC (I),CPEAK (I),CPXX (I),CPYY (I),CPXY (I)
28     27  FORMAT (3F10.4,3E13.7)
29     KCREP (I)=-2147483600
30     60  CONTINUE
31     READ (8,37) LSTC
32     READ (8,37) NC
33     READ (8,7) NCTOT
34     READ (8,37) (LISTC (I),I=1,NCTOT)
35     65  DO 100 I=1,NHILLS
36     SX (I)=SX (I)*1.625
37     SY (I)=SY (I)*1.625
38     XC (I)=(XC (I)-500.)*100.
39     YC (I)=(YC (I)-930.)*100.
40     TWO RHO (I)=2.*RHO (I)
41     SCALE (I)=-1./ (2.*(1.-RHO (I)*2))
42     TWO SCL (I)=2.*SCALE (I)
43     KHREP (I)=-2147483600
44     C  ALL VALUES NOW IN METERS ON 0 -- 10,000 GRID
45     100 CONTINUE
46     KTREP=-2147483600
47     KH=0
48     KHW=0
49     KV=0
50     KN=0

```



```
51      KGRS=0
52      KELL=0
53      KINT=0
54      RETURN
55      END
```



```

1      C
2      SUBROUTINE ROUTE
3      C
4      COMMON /GRP3/ NBU,NRU,FL(6),FO(6),NOI(3),XIC(3,200),YIC(3,200),
5      IDIR(3,200),AVSP,ISPD
6      I, IUSTAT(6),II(6),LOST(6,6),VISFRA,VISFRB,SIZETK,
7      ISIZETW,NT(6),NF(6),SRF,DISHAX,
8      INLOSC(6,6),VISFR(6,6),RMINTK,RMXTK,RMINTW,RMXTW,OP,TOWFR,TNKFR,
9      IPTT(3,3),RF,POR(6,6),APOR(6,6),LOR(6,6),NA(6),OFL(6),POL(6)
10     DIMENSION XLQC(3,20),YLQC(3,20),N(3)
11     IF (ISPD.EQ.4) DST=80.463
12     IF (ISPD.EQ.3) DST=67.053
13     IF (ISPD.EQ.2) DST=59.643
14     IF (ISPD.EQ.1) DST=40.232
15     DO 300 I=1,NRU
16     READ(5,15) N(I)
17     15 FORMAT(12)
18     NL=N(I)+1
19     DO 200 IN=2,NL
20     READ(5,201) XLQCS,YLQCS
21     201 FORMAT(F6.1,1X,F6.1)
22     XLQC(I,IN)=XLQCS
23     YLQC(I,IN)=YLQCS
24     200 CONTINUE
25     XLQC(I,1)=XIC(I,1)
26     YLQC(I,1)=YIC(I,1)
27     IDIR(I,1)=0
28     NL=N(I)
29     NUM=2
30     DO 305 J=1,NL
31     XL=XLQC(I,J+1)-XLQC(I,J)
32     YL=YLQC(I,J+1)-YLQC(I,J)
33     DIST=SQRT(XL**2+YL**2)
34     Y=ABS(YL)
35     Z=Y/XL
36     ANGL=ATAN(Z)
37     DEG=ANGL*57.2958
38     IF (J.EQ.1) GO TO 320
39     XLN=(DIST-EXTRA)*COS(ANGL)
40     DIST=(DIST+EXTRA)-DST
41     YLN=(DIST-EXTRA)*SIN(ANGL)
42     XIC(I,NUM)=XIC(I,NUM-1)+XLN+XLE
43     IF (YL.GT.0.) GO TO 325
44     YLN=-YLN
45     325 YIC(I,NUM)=YIC(I,NUM-1)+YLN+YLE
46     IF (YL.GT.0.) GO TO 340
47     IDIR(I,NUM)=-IFIX(DEG)
48     GO TO 341
49     340 IDIR(I,NUM)=IFIX(DEG)
50     341 NUM=NUM+1

```



```

51      320 XLN=DST*COS (ANGL)
52      YLN=DST*SIN (ANGL)
53      IF (YL.GT.O.) GO TO 310
54      YLN=-YLN
55      310 IF (DIST.LT.DST) GO TO 315
56      XIC (I,NUM)=XIC (I,NUM-1)+XLN
57      YIC (I,NUM)=YIC (I,NUM-1)+YLN
58      IF (YL.GT.O.) GO TO 342
59      IDIR (I,NUM)=-IFIX (DEG)
60      GO TO 343
61      342 IDIR (I,NUM)=IFIX (DEG)
62      343 DIST=DIST-DST
63      NUM=NUM+1
64      GO TO 310
65      315 EXTRA=DIST
66      XLE=EXTRA*COS (ANGL)
67      YLE=EXTRA*SIN (ANGL)
68      IF (YL.GT.O.) GO TO 305
69      YLE=-YLE
70      305 CONTINUE
71      300 CONTINUE
72      RETURN
73      END

```



```

1      C
2      SUBROUTINE LAMOR (I,J,PCTVIS,OETRAT,PK)
3      C
4      C      SUBROUTINE TO COMPUTE DETECTION RATE (OETRAT) OF TARGET J
5      C      BY OBSERVER I GIVEN THE VISIBLE FRACTION (PCTVIS).
6      C
7      COMMON /GRP1/ IPROIR (6) , ISECWD (6) , MVTDIR (6) , X (6) , Y (6) , SPD (6)
8      TCFAC=1.0
9      ZERO=0.00001
10     PAI=22.0/7.0
11     7 D= (ISECWD (I) *PAI/180.0) /2.0
12     BBB= (1.0/ (2.0* (SIN (D) -D*COS (D) ) ) )
13     IF (ABS (BBB) .LT. ZERO) BBB=0.0
14     AAA= (-BBB) *COS (D)
15     IF (ABS (AAA) .LT. ZERO) AAA=0.0
16     OTANG=ATAN2 ((Y (J) -Y (I)) , (X (J) -X (I) ) )
17     PO=IPROIR (I) *PAI/180.0
18     IF ((PD*OTANG) .GE.0.0) GOTO 1
19     IF (PD.LT.0.0) GOTO 9
20     ANGLE=2*PAI+OTANG-PD
21     GOTO 10
22     9 ANGLE=2*PAI+PO-OTANG
23     10 IF (ANGLE.GT.PAI) ANGLE=2*PAI-ANGLE
24     GOTO 2
25     1 ANGLE=ABS (PO-OTANG)
26     2 IF (ANGLE.GT.D) GOTO 3
27     OUP=PD+D
28     DLOW=PD-D
29     ANGLFT=OTANG+ (15.0*PAI/180.)
30     IF (ANGLFT.GT.OUP) ANGLFT=OUP
31     ANGLAT=OTANG- (15.*PAI/180.)
32     IF (ANGLAT.LT.DLOW) ANGLAT=DLOW
33     PK=BBB* (SIN (ANGLFT) -SIN (ANGLAT) ) +AAA* (ANGLFT-ANGLAT)
34     IF (PK.LT.0.0) GOTO 3
35     IF (PK.GT.1.0) GOTO 5
36     GOTO 8
37     3 PK=0.0
38     OETRAT=0.0
39     GOTO 6
40     5 PK=1.0
41     8 RANGE=SQRT ((X (J) -X (I) ) **2+ (Y (J) -Y (I) ) **2)
42     RR=0.001*RANGE/PCTVIS
43     TOANG=ATAN2 ((Y (I) -Y (J) ) , (X (I) -X (J) ) )
44     AO=MVTDIR (J) *PAI/180.0
45     HORVEL=ABS (SPD (J) *SIN (TOANG-AO) )
46     DENOM=1.453+TCFACT* (0.5978+2.188* (RR**2) -0.5038*HORVEL)
47     IF (DENOM.LE.ZERO) OENOM=ZERO
48     DETRAT=0.003+1.088/OENOM
49     DETRAT=OETRAT*PK
50     6 RETURN

```



```

1      SUBROUTINE ELEV(X,Y,THAC)
2      C
3      COMMON /HILLS/ XC(100),YC(100),PEAK(100),SX(100),SY(100),RHO(100)
4      COMMON /HILLS/ SCALE(100),TWRHO(100),TWSCL(100),BASE
5      COMMON /HILLS/ NHILLS
6      COMMON /GRID/ LST(10,10),NHL(10,10),LISTH(450),KHREP(100),KTREP
7      COMMON /GRID/ LSTC(10,10),NC(10,10),LISTC(400),KCREP(150)
8      DATA NGRID/10/,GSIZE/1000./
9      C FUNCTION TO COMPUTE TERRAIN ELEVATION FOR GIVEN X, Y COORDINATES,
10     ZMAX=BASE
11     IX=1+IFIX(X/GSIZE)
12     IF (IX.GT.NGRID) IX=NGRID
13     IY=1+IFIX(Y/GSIZE)
14     IF (IY.GT.NGRID) IY=NGRID
15     IF (NHL(IX,IY).EQ.0) GO TO 150
16     LS=LST(IX,IY)
17     LEND=LS+NHL(IX,IY)-1
18     DO 100 L=LS,LEND
19     I=LISTH(L)
20     QX=(X-XC(I))/SX(I)
21     QXSQ=QX*QX
22     IF (QXSQ.GE.9.) GO TO 100
23     QY=(Y-YC(I))/SY(I)
24     QYSQ=QY*QY
25     IF (QYSQ.GE.9.) GO TO 100
26     QXY=TWRHO(I)*QX*QY
27     FACTOR=SCALE(I)*(QXSQ+QYSQ+QXY)
28     IF (FACTOR.LT.-3.) GO TO 100
29     HT=PEAK(I)*EXP(FACTOR)
30     IF (HT.LE.ZMAX) GO TO 100
31     ZMAX = HT
32     100 CONTINUE
33     150 THAC=ZMAX
34     RETURN
35     END

```



```

1      SUBROUTINE STOCH(I,RANGE,A)
2      C
3      C      SUBROUTINE TO COMPUTE STOCHASTIC ATTRITION COEFFICIENT
4      C
5      COMMON /GRP6/ ALPHA(6)
6      COMMON /GRP3/ NBU,NRU,FL(6),FO(6),NOI(3),XIC(3,200),YIC(3,200),
7      IIDIR(3,200),AVSP,ISPD
8      I, IUSTAT(6),II(6),LOST(6,6),VISFRA,VISFRB,SIZEK,
9      ISIZETW,NT(6),NF(6),SRF,DISMAX,
10     INLOSC(6,6),VISFR(6,6),RMINTK,RMXTK,RMINTW,RMXTW,OP,TOWFR,TNKFR,
11     IPTT(3,3),RF,POA(6,6),APOA(6,6),LOA(6,6),NA(6),OFL(6),POL(6)
12     IF (I.EQ.2) GO TO 5003
13     A=ALPHA(I)*((1.0-RANGE/RMXTW)**2)
14     GO TO 5004
15 5003 A=ALPHA(I)*((1.0-RANGE/RMXTK)**2)
16 5004 RETURN
17     END

```



```

1      SUBROUTINE ETK(I,RANGE,T)
2      C
3      C      SUBROUTINE TO COMPUTE EXPECTED TIME TO KILL A TARGET.
4      C
5      COMMON /GRP2/ TA(2),T1(2),TH(2),TM(2),TF1(2),TF2(2),TF3(2),
6      IP(2,6),PHH(2,6),PHM(2,6),PKH(2,6),TF(2)
7      IF(I.EQ.2) GOTO 5
8      TF(I)=TF1(I)
9      GOTO 6
10     5 IF(RANGE.GT.1000.0) GOTO 7
11       TF(I)=TF1(I)-(TF1(I)*(1000.0-RANGE)/1000.0)
12       GOTO 6
13     7 IF(RANGE.GT.2000.0) GOTO 8
14       TF(I)=TF2(I)-((TF2(I)-TF1(I))*(2000.0-RANGE)/1000.0)
15       GOTO 6
16     8 TF(I)=TF3(I)-((TF3(I)-TF2(I))*(3000.0-RANGE)/1000.0)
17     6 J=(RANGE+250.0)/500.0
18       IF(J.GT.6) J=6
19       T=TA(I)+T1(I)-TH(I)+((TH(I)+TF(I))/PKH(I,J))+((TM(I)+TF(I))/
20       PHH(I,J))*((1.0-PHH(I,J))/PKH(I,J)+PHH(I,J)-P(I,J))
21       RETURN
22     END

```



```

1      C
2      SUBROUTINE SORT (I,M)
3      COMMON /GRPS/ LOT (6,6),ROT (6,6)
4      DO 19 J=1,M
5      IF (ROT (I,M).GE.ROT (I,J)) GOTO 19
6      21 R=ROT (I,J)
7      NN=LOT (I,J)
8      ROT (I,J)=ROT (I,M)
9      LOT (I,J)=LOT (I,M)
10     ROT (I,M)=R
11     LOT (I,M)=NN
12     19 CONTINUE
13     RETURN
14     END

```



```

1      SUBROUTINE KOVER (Z0,TMACT,SIZET,ZT,S,HTS,ZS,VISFAT)
2      C
3      IF (S.EQ.0.) GO TO 2000
4      IF (HTS.GE.ZS) GO TO 2050
5      HEXT=Z0+(HTS-Z0)/S
6      EVIST=AMAX1(HEXT,TMACT)
7      IF (EVIST.GE.ZT) GO TO 2050
8      IF (EVIST.LE.ZT-SIZET) RETURN
9      VIS=(ZT-EVIST)/SIZET
10     IF (VIS.LT.VISFAT) VISFAT=VIS
11     RETURN
12 2000 IF (HTS.LT.Z0) RETURN
13 2050 VISFAT=0.0
14     RETURN
15     END

```


APPENDIX C

Definition of Variables in Computer Program

ALPHA(I)	= Initial attrition-rate coefficient for stochastic attrition module.
APOA(I,J)	= The average proportion of the j^{th} attacker of unit i allocated to fire on unit i.
AVSP	= Average speed of moving attacking units.
BREAK	= Breakpoint distance between attackers and defenders.
DISMAX	= Maximum distance allowed between attacking units before the leading unit is delayed.
DIST	= The straight-line distance between two movement nodes inputed by the user.
DST	= The distance to be moved each time step by attacking units.
FL(I)	= Force level of unit i.
FO(I)	= Initial force level of unit i.
IALT	= Denotes whether the user desires alternate defensive positions or not.
IC	= Counts number of time units a defender has been moving.
IDIR(I,J)	= Direction of j^{th} interval in i^{th} route.
II(I)	= Interval index for unit i.
IMOVE	= Number of time units a defender is allowed for moving to an alternate position.
IPRDIR(I)	= Primary direction of movement for unit i.
IRTE	= Denotes whether user wants to input routes or not.
IS	= Random number seed used for stochastic attrition.
ISECWD(I)	= Width of search sector for unit i.

ISPD = Input variable to denote user's desired speed for attackers movement.

ITEM = Input variable denoting number of time steps allowed for defender's move.

ITIME = Current time, in seconds, of battle.

ITRIT = Input variable denoting whether attrition will be stochastic or deterministic.

IUSTAT(I) = Current status of unit i.

LCA(I,J) = The number of the j^{th} attacker of unit i.

LOST(I,J) = Denotes whether line-of-sight exists between unit i and j or not.

LOT(I,J) = The number of the j^{th} target of unit i.

MVTDIR(I) = Movement direction of unit i.

N(I) = Number of nodes inputed by user for route i.

NA(I) = Number of attackers of unit i.

NBU = Number of blue units.

NF(I) = Number of time units unit i is allowed to fire at the same location.

NLOSC(I,J) = Number of continuous time steps that line-of-sight does not exist between unit i and unit j.

NOI(I) = Number of intervals in the i^{th} route.

NRU = Number of Red Units.

NT(I) = Number of targets of unit i.

OFL(I) = Force level of unit i during previous time step.

P(I,J) = Probability of 1^{st} round hit by unit i in range band j.

PHH(I,J) = Probability of a hit following a hit by unit i in range band j.

PHM(I,J) = Probability of a hit following a miss by unit i in range band j.

PKH(I,J) = Probability of a kill given a hit by unit i in range band j.

PM = The proportion of time a moving unit is searching for targets.

POA(I,J) = The proportion of the j^{th} attacker of unit i allocated to fire on unit i.

POL(I) = Percent of unit i lost during the current time step.

PTT(I,J) = Proportion of surviving firepower allocated to the i^{th} target if there are j targets available.

RANGE = Current minimum distance between attackers and defenders.

RMINTK = Minimum effective range for attacking weapon system.

RMINTW = Minimum effective range for defending weapon system.

RMXTK = Maximum effective range for attacking weapon system.

RMXTW = Maximum effective range for defending weapon system.

ROT(I,J) = The range of the j^{th} target of unit i.

SIZETK = Size of attacking vehicle.

SIZETW = Size of defending vehicle.

TA(K) = Time to acquire a target for k^{th} weapon system type ($k = 1,2$).

TF1(K) = Time of flight to 1000m for k^{th} weapon system type ($k = 1,2$).

TF2(K) = Time of flight to 2000m for k^{th} weapon system type ($k = 1,2$).

TF3(K) = Time of flight to 3000m for k^{th} weapon system type ($k = 1,2$).

TH(K) = Time to fire a round following a hit
 for weapon system type k (k = 1,2).

TI(K) = Time to fire first round after target
 has been acquired for weapon system
 type k (k = 1,2).

TM(K) = Time to fire a round following a miss
 for weapon system type k (k = 1,2).

TNKFR = Firing rate for attacking weapon system.

TOWFR = Firing rate for defending weapon system.

TPOL(I) = Total percentage of lost since battle
 began for unit i.

VISFR(I,J) = The fraction of unit i seen by unit j.

VISFRA = Fraction of unit A as seen by unit B.

VISFRB = Fraction of unit B as seen by unit A.

X(I),Y(I) = Coordinates of unit i.

XA(I),YA(I) = Coordinates of alternate position for
 defender i.

XIC(I,J) = Coordinates of the j^{th} interval endpoint
 YIC(I,J) of the route for unit i.

XL,YL = Distance added to previous interval
 endpoint for vehicle to move DST during
 a time step.

XLOC(I,J) = Coordinates of the j^{th} node inputed by
 YLOC(I,J) the user for the route of unit i.

BIBLIOGRAPHY

Bonder, S., "The Lanchester Attrition-Rate Coefficient," Operations Research, v. 15, p. 221-232, May 1967.

Bonder, S., "The Mean Lanchester Attrition Rate," Operations Research, v. 18, p. 179-181, June 1970.

Clark, G. M., The Combat Analysis Model, Ph.D. Thesis, Ohio State University, Columbus, Ohio, 1969.

Karr, A. F., Stochastic Attrition Models of Lanchester Type, P-1031, Institute for Defense Analysis, Arlington, Virginia, June 1974.

Smoler, J., Operational Lanchester-Type Model of Small Unit Land Combat, M.S. Thesis, Naval Postgraduate School, Monterey, California, 1979.

Taylor, J. G., Force-on-Force Attrition Modelling, p. 20-89, Military Applications Section of Operations Research Society of America, January 1980.

Taylor, J. G., "Solving Lanchester-Type Equations for 'Modern Warfare' with Variable Coefficients," Operations Research, v. 22, p. 756-770, October 1974.

Wallace, W. S., and Hagewood, E. G., Simulation of Tactical Alternative Responses (STAR), M. S. Thesis, Naval Postgraduate School, Monterey, California, December 1978.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 55 Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
4. Professor James G. Taylor, Code 55Tw Department of Operations Research Naval Postgraduate School Monterey, California 93940	10
5. Professor Glenn F. Lindsay, Code 55Ls Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
6. Professor Samuel H. Parry, Code 55Py Department of Operations Research Naval Postgraduate School Monterey, California 93940	1
7. Captain Glenn M. Mills 710 Manhasset Road Charlotte, North Carolina 28209	2
8. Command and General Staff College Attn: Education Advisor Room 123, Bell Hall Ft. Leavenworth, Kansas 66027	1
9. Headquarters, Department of the Army MILPERCEN Attn: DAPC-OPF-C 200 Stovall Street Alexandria, Virginia 22332	1

Thesis
M595 Mills 189349
c.1 The enrichment of
Smoler's model of land
combat.

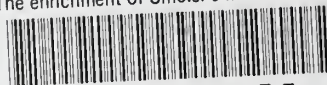
19 OCT 82
24

27109
27810
2807

Thesis
M595 Mills 189349
c.1 The enrichment of
Smoler's model of land
combat.

thesM595

The enrichment of Smoler's model of land



3 2768 000 98435 5

DUDLEY KNOX LIBRARY